NAVAL POSTGRADUATE SCHOOL
Monterey, California

2

AD-A272 967

DTIC
ELECTE
NOV 2 2 1993
S A

THESIS

COMPARISON OF HIGHER ORDER MOMENT SPECTRUM ESTIMATION TECHNIQUES

by

Jeffrey F. McAloon

September, 1993

Co-Advisor:	Ralph Hippenstiel
Co-Advisor:	Monique P. Fargues

Approved for public release; distribution is unlimited.

93-28380

93 11 19 050

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release;
distribution is unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Electrical Engineering Dept.
Naval Postgraduate School

6b. OFFICE SYMBOL *(if applicable)*
EC

7a. NAME OF MONITORING ORGANIZATION
Naval Postgraduate School

6c. ADDRESS *(City, State, and ZIP Code)*
Monterey, CA 93943-5000

7b. ADDRESS *(City, State, and ZIP Code)*
Monterey, CA 93943-5000

8a. NAME OF FUNDING/SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(if applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
COMPARISON OF HIGHER ORDER MOMENT SPECTRUM ESTIMATION TECHNIQUES (U)

12. PERSONAL AUTHOR(S)
McAloon, Jeffrey F.

13a. TYPE OF REPORT
Master's Thesis

13b. TIME COVERED
FROM TO

14. DATE OF REPORT *(Year, Month, Day)*
September 1993

15. PAGE COUNT
124

16. SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
bispectrum; 1-1/2 D; IPS; higher-order moments; cumulants

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

This thesis compares the detection performance of the 1-1/2 D instantaneous power spectrum (1-1/2 D_{ips}), the bispectrum, the instantaneous higher-order moment slice (IHOMS) method, and the spectrogram for multi-component stationary signals, harmonically related stationary signals, and multi-component linear FM signals corrupted by additive white Gaussian noise. In addition, a determination of the relative processing gain between the 1-1/2 D_{ips} method and the spectrogram is made for stationary signals in noise.

The results of this thesis show that 1-1/2 D_{ips} has a processing gain advantage over that of the spectrogram for a range of input SNR that depends upon the size of the data window. Under some conditions, the bispectrum can detect both harmonic coupling and phase coupling between the components of multi-component signals. IHOMS' ability to detect linear chirps in noise is limited to chirps having different slew rates, and the method has a significantly greater computational cost than both the spectrogram and 1-1/2 D_{ips}.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED □ SAME AS RPT. □ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
HIPPENSTIEL, R. / FARGUES, M.

22b. TELEPHONE *(Include Area Code)*
(408) 656-2633/2859

22c. OFFICE SYMBOL
EC/Hi EC/Fa

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

Approved for public release; distribution is unlimited.

COMPARISON OF HIGHER ORDER MOMENT SPECTRUM ESTIMATION TECHNIQUES

by

Jeffrey F. McAloon
Lieutenant, United States Navy
B.S.E.E., University of South Florida, 1983

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1993

Author: J F McAloon
Jeffrey F. McAloon

Approved By: Ralph Hippenstiel
Ralph Hippenstiel, Co-Advisor

Monique P. Fargues, Co-Advisor

R. Clark Robertson
for Michael A. Morgan, Chairman,
Department of Electrical and Computer Engineering

ABSTRACT

This thesis compareݍ the detection performance of the 1-1/2 D instantaneous power spectrum (1-1/2 D_{ips}), the bispectrum, the instantaneous higher-order moment slice (IHOMS) method, and the spectrogram for multi-component stationary signals, harmonically related stationary signals, and multi-component linear FM signals corrupted by additive white Gaussian noise. In addition, a determination of the relative processing gain between the 1-1/2 D_{ips} method and the spectrogram is made for stationary signals in noise.

The results of this thesis show that 1-1/2 D_{ips} has a processing gain advantage over that of the spectrogram for a range of input SNR that depends upon the size of the data window. Under some conditions, the bispectrum can detect both harmonic coupling and phase coupling between the components of multi-component signals. IHOMS' ability to detect linear chirps in noise is limited to chirps having different slew rates, and the method has a significantly greater computational cost than both the spectrogram and 1-1/2 D_{ips}.

TABLE OF CONTENTS

Accesion For

NTIS CRA&I ☒
DTIC TAB □
Unannounced □
Justification

By
Distribution/

Availability Codes

Dist | Avail and/or Special

A-1

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. OVERVIEW

Spectral estimation is an important data analysis tool used in signal processing to determine the dis~~r~~ribution of power as a function of frequency. The "classical" Fourier transform methods and a variety of more recently developed parametric methods are now commonly used to estimate power spectral density. These methods are based on the autocorrelation domain and therefore rely upon information contained in the second order moment of the data sequence. Recent improvements in computational capability have opened up a new and active research area concerned with the extraction of additional information contained in the data sequence's higher-than-second order statistics. [Refs. 1,2]

Higher order statistics involve cumulants rather than moments. Cumulants and moments are similar, and each can be expressed in terms of the other. In fact, under certain conditions they are identical. Higher order spectra (HOS), also known as polyspectra, are formed by taking the Fourier transform of the cumulant sequence. For example, the two-dimensional Fourier transform of the third order cumulant yields the polyspectrum known as the bispectrum. Likewise, the three-dimensional Fourier transform of the fourth order cumulant is called the trispectrum. If the first moment is zero, the second order cumulant is equal to the second order moment, and the one-dimensional Fourier transform of either sequence produces the familiar power spectrum. [Refs. 1,3]

Higher order statistics and their spectra possess some exploitable properties that their second order counterparts do not. One potential advantage involves a greater resistance to the effects of certain types of additive noise. For white or colored, zero mean Gaussian processes, the cumulant of any order greater than two is zero and the corresponding polyspectrum is zero. Consequently, polyspectra are expected to be more resistant than the power spectrum is to the effects of additive Gaussian noise. One limitation of the power spectrum is that phase cannot be accurately reconstructed unless the underlying signal or system is minimum phase. In contrast, higher order spectra preserve true phase information. Finally, HOS are useful for detection and classification of non-linearities. Depending upon the application, the extra computational cost of HOS may be justified. [Refs. 3,4]

The bispectrum, the 1-1/2 D instantaneous power spectrum [Ref. 5], and the instantaneous higher order moment slice (IHOMS) method [Ref. 6] are studied in this thesis. Performance comparisons are made with respect to each method as well as to an accepted second order method.

B. THESIS OUTLINE

The following describes the organization of the remainder of this thesis. Chapter II first defines cumulants and shows their relation to moments. The essentials are then presented for each spectral estimation method studied. Chapter III makes a processing gain comparison between the 1-1/2 D instantaneous power spectrum and the spectrogram. Chapter IV displays simulation results that show how the different methods perform on a variety of signals. Chapter V presents conclusions and suggestions for future work. Appendices contain the detailed steps of Chapter III calculations and the Matlab programs used to execute the computer simulations.

II. ELEMENTS OF HIGHER ORDER SPECTRAL ANLYSIS

This chapter describes the higher order based techniques compared later in this thesis. Sufficient information is presented in order to develop an adequate understanding of each method. We first define cumulants and show how they are used instead of moments to form polyspectra.

A. CUMULANTS AND MOMENTS

In general, the characteristic function, $\Phi(j\omega)$, is defined as the conjugated Fourier transform of a random variable's probability density function, f(x) [Ref. 7]:

$$\Phi(j\omega) = \int_{-\infty}^{\infty} f(x)\, exp\,(j\omega x)\, dx. \tag{2.1}$$

The right-hand side of (2.1) is simply the expectation of $exp\,(j\omega X)$. The n^th^ moment of the random variable X can be found by evaluating the n^th^ derivative of the characteristic function with ω set equal to zero [Ref. 7]:

$$E\{X^n\} = (-j)^n \frac{d^n}{d\omega^n} \Phi(j\omega) \bigg|_{\omega = 0} . \tag{2.2}$$

In comparison, the n^th^ cumulant is defined as the nth derivative of the natural logarithm of the characteristic function evaluated at $\omega = 0$ [Ref. 4]:

$$C\{X^n\} = (-j)^n \frac{d^n}{d\omega^n} ln\,[\Phi(\omega)] \bigg|_{\omega = 0} . \tag{2.3}$$

Although (2.3) clearly shows the basic difference between a moment and a cumulant, it does not readily show that a cumulant can be considered as a moment with its lower order statistical dependence removed [Ref. 8]. This is perhaps best seen by computing cumulants in an alternate manner that involves a partitioning of equal and lower order moments.

Given a set of random variables $X = \{X_1, X_2, \ldots, X_k\}$, $I_X = \{1, 2, \ldots, k\}$ is defined to be the set of component indices contained in X. Denoting a subset of I_X by I, using $m_X(I)$ to represent the moment of those components of X comprising X_I, and denoting the cumulant of X_I as $c_X(I)$, the moment-to-cumulant (M-C) equation can be written as [Ref. 3]:

$$c_X(I) = \sum_{\bigcup_{p=1}^{q} I_p = I} (-1)^{q-1}(q-1)! \prod_{p=1}^{q} m_X(I_p); \tag{2.4}$$

where I_p are non-intersecting non-empty subsets of I that form the partitions, and q is varied from one to the number of elements in I. The summation in (2.4) is over all unique partitions and for all q. The first order cumulant is equal to the first order moment since only one unique partition exists for one random variable. Table 2.1 shows how (2.4) is applied to compute the moment partitions that form the second order cumulant. Table 2.2 displays the details of the third order cumulant calculation. Summing the last column of Table 2.1 yields the second order cumulant. Summing the last column of Table 2.2 produces the third order cumulant. The first three orders of cumulants, expressed in terms of the component indices of a set of random variables, are then:

$$\begin{aligned} c(1) &= m(1), \\ c(1,2) &= m(1,2) - m(1)m(2), \\ c(1,2,3) &= m(1,2,3) + 2m(1)m(2)m(3) - m(1,2)m(3) \\ &\quad - m(1,3)m(2) - m(2,3)m(1). \end{aligned}$$

Calculation of the fourth order cumulant is similarly detailed in [Ref. 3]. An expression that calculates moments from partitioned cumulants is also given in [Ref. 3].

TABLE 2.1: COMPUTATION OF SECOND ORDER CUMULANT

I_1	I_2	q	M-C equation (2.4)
1,2		1	$m(1,2)$
1	2	2	$-m(1)m(2)$

TABLE 2.2: COMPUTATION OF THIRD ORDER CUMULANT

I_1	I_2	I_3	q	M-C equation (2.4)
1,2,3			1	m(1,2,3)
1,2	3		2	-m(1,2)m(3)
1,3	2		2	-m(1,3)m(2)
2,3	1		2	-m(2,3)m(1)
1	2	3	3	2m(1)m(2)m(3)

Inspection of the moment-cumulant expressions show that dependencies among random variables are removed when cumulants are calculated. In fact, both the second and third order cumulants are zero if all the random variables are independent of each other. [Ref. 8]

Another observation regarding the above moment-cumulant expressions applies in the commonly encountered situation where the means of the random variables are equal to zero. When this is true, the first order cumulant is zero since it equals the mean, the second order cumulant simply equals the variance, and the third order cumulant equals the third order moment. [Ref. 3]

There is more than one acceptable choice when it comes to choosing which term(s) to conjugate in the expectation expression for moments of order greater than two. Different choices have different consequences when the signal being processed is complex [Refs. 9, 10]. One consequence is considered later when the symmetry region of the bispectral plane is discussed. For now, the conjugated terms are chosen arbitrarily. The following expressions show the details of the moment calculations in the last set of equations for the zero mean case [Ref. 11]:

$$C^{(1)} = E\{x[n]\} = 0, \tag{2.5}$$

$$C^{(2)}[l_1] = E\{x^*[n]x[n+l_1]\}, \tag{2.6}$$

$$C^{(3)}[l_1, l_2] = E\{x^*[n]x[n+l_1]x[n+l_2]\}. \tag{2.7}$$

The expression for the fourth order cumulant as calculated by (2.4) has 15 moment terms. If the mean is zero all but four terms go to zero when the signal is real [Refs. 3,11]:

$$\begin{aligned} C^{(4)}[l_1, l_2, l_3] = {} & E\{x[n]x[n+l_1]x[n+l_2]x[n+l_3]\} \\ & -E\{x[n]x[n+l_1]\}E\{x[n+l_2]x[n+l_3]\} \\ & -E\{x[n]x[n+l_2]\}E\{x[n+l_1]x[n+l_3]\} \\ & -E\{x[n]x[n+l_3]\}E\{x[n+l_1]x[n+l_2]\}. \end{aligned} \tag{2.8}$$

Provided that two of the four random variables are conjugated in the expectation terms of (2.8), the fourth order cumulant of a zero mean complex process has just three expectation terms. One of the last three terms in (2.8) will be zero. The zero term depends upon which variables are conjugated. [Ref. 11]

Cumulants have three properties that make them more desirable than moments when it comes to higher order statistics:

1. Each cumulant is independent of all lower order cumulants. Consequently, all cumulants of order greater than two are equal to zero for a Gaussian process, as a Gaussian process is completely characterized by its first and second

moments. Higher order moments, on the other hand, can contain information about lower order moments. In fact, only higher order moments of odd order are zero for a Gaussian process. The non-zero even order higher moments do not contain any information not already contained in the first two moments. [Ref. 8]

2. If a set of n random variables can be divided into more than one statistically independent subset, then the nth order cumulant, unlike the nth order moment, is equal to zero. [Ref. 4]
3. Unlike moments, the cumulant of the sum of two independent stationary random processes is equal to the sum of the cumulants of each process. [Ref. 4]

B. THE BISPECTRUM

1. Definition and Region of Support

The bispectrum is defined as the two-dimensional Fourier transform of the third order cumulant [Ref. 4]:

$$B(\omega_1, \omega_2) = \frac{1}{(2\pi)^2} \sum_{l_1 = -\infty}^{\infty} \sum_{l_2 = -\infty}^{\infty} C^{(3)}(l_1, l_2)\, exp\{-j(\omega_1 l_1 + \omega_2 l_2)\}\,. \qquad (2.9)$$

Analogous to the power spectrum, the bispectrum can also be defined with frequency domain quantities Given N samples of stationary signal $x(n)$, its Fourier transform is

$$X(\omega) = \sum_{n=0}^{N-1} x(n)\, exp(-j\omega n)\,. \qquad (2.10)$$

Assuming that the third order cumulant in (2.9) is computed with the conjugation scheme shown in (2.7), the equivalent frequency domain expression for the bispectrum is obtained through an extension of the periodogram [Refs. 9,12]:

$$B(\omega_1, \omega_2) = \frac{1}{N} X(\omega_1) X(\omega_2) X^* (\omega_1 + \omega_2) ; \qquad (2.11)$$

where $|\omega_1| \leq \pi$, $|\omega_2| \leq \pi$, and $|\omega_1 + \omega_2| \leq \pi$. Equations (2.9) and (2.11) represent two different non-parametric methods that can be used to calculate the bispectrum. The approach taken by (2.9) is called the indirect method, while the approach used by (2.11) is known as the direct method. [Ref. 1]

In general, the bispectrum's region of support is a hexagon centered at the origin of the (ω_1, ω_2) bi-frequency plane. Evaluation of the mean product of three Fourier amplitude terms shows that the bispectral plane exhibits certain symmetries. For a stationary, real, continuous time signal the expected value of the product of three Fourier components is [Ref. 13]:

$$E\{X_c(\omega_1) X_c(\omega_2) X_c(\omega_3)\} = B_c(\omega_1, \omega_2) \delta(\omega_1 + \omega_2 + \omega_3) ; \qquad (2.12)$$

where the subscript c denotes continuous time quantities.

Three properties of (2.12) determine bispectral symmetry. First, since the frequency indexes must sum to zero, $\omega_3 = -\omega_1 - \omega_2$. Second, the frequency indexes in the expectation operation can be interchanged. Third, for the bispectrum of a real signal, conjugate symmetry results since $B(-\omega_1, -\omega_2) = B^* (\omega_1, \omega_2)$. Using the first property to express ω_3 in terms of ω_1 and ω_2, the symmetry lines for a continuous time signal are [Ref. 13]:

$\omega_1 = \omega_2$,

$2\omega_1 = -\omega_2$ (from $\omega_1 = \omega_3$),

$2\omega_2 = -\omega_1$ (from $\omega_2 = \omega_3$),

$\omega_1 = -\omega_2$,

$\omega_2 = 0$ (from $\omega_1 = -\omega_3$),

$$\omega_1 = 0 \qquad \text{(from } \omega_2 = -\omega_3\text{)}.$$

If the signal is band-limited to ω_o and sampled at a rate of $\omega_s = 2\omega_o = \frac{2\pi}{\tau}$, the bispectrum term in (2.12) becomes $B(\omega_1 + \frac{2\pi n}{\tau}, \omega_2 + \frac{2\pi n}{\tau})$ after applying the approximation:

$$X(\omega) \cong \sum_{n=-\infty}^{\infty} X_c(\omega + \frac{2\pi n}{\tau}). \tag{2.13}$$

The frequency indexes in (2.12) now must sum to $\frac{2\pi n}{\tau}$ instead of zero. As a result, the following set of symmetry lines are formed in the bispectrum of a sampled signal [Ref. 13]:

$$\omega_1 = \omega_2,$$

$$2\omega_1 + \omega_2 = \frac{2\pi n}{\tau},$$

$$\omega_1 + 2\omega_2 = \frac{2\pi n}{\tau},$$

$$\omega_1 = -\omega_2,$$

$$\omega_2 = \frac{2\pi n}{\tau},$$

$$\omega_1 = \frac{2\pi n}{\tau}.$$

Figure 2.1 shows the region of support and its symmetry lines for the first of an infinite number of bispectral hexagons given a sampled real signal [Refs. 12,13]. The sampling period is assumed to be equal to one. A complex signal has the same hexagonal support region as that of a real signal but it is symmetrical about only one of three dotted symmetry lines shown in Figure 2.1. Depending upon the term conjugated in (2.7) used to compute the third order cumulant for the indirect method, or which term in (2.11) is conjugated to compute the bispectrum by the direct method, the bispectrum of the complex signal exhibits symmetry in only one quadrant about either the $\omega_1 = \omega_2$ line, the $\omega_2 = -2\omega_1$ line, or the $\omega_1 = -2\omega_2$ line. The conjugation scheme used exclusively in this study is shown in (2.7) and (2.11). Under this scheme, the bispectrum of a complex signal displays two-fold symmetry in the first quadrant about the $\omega_1 = \omega_2$ line. There are two other possible ways to conjugate just one term, and three possible ways to conjugate any two terms in the applicable expressions. The bispectrum symmetry for these other schemes are considered fully in [Ref. 9].

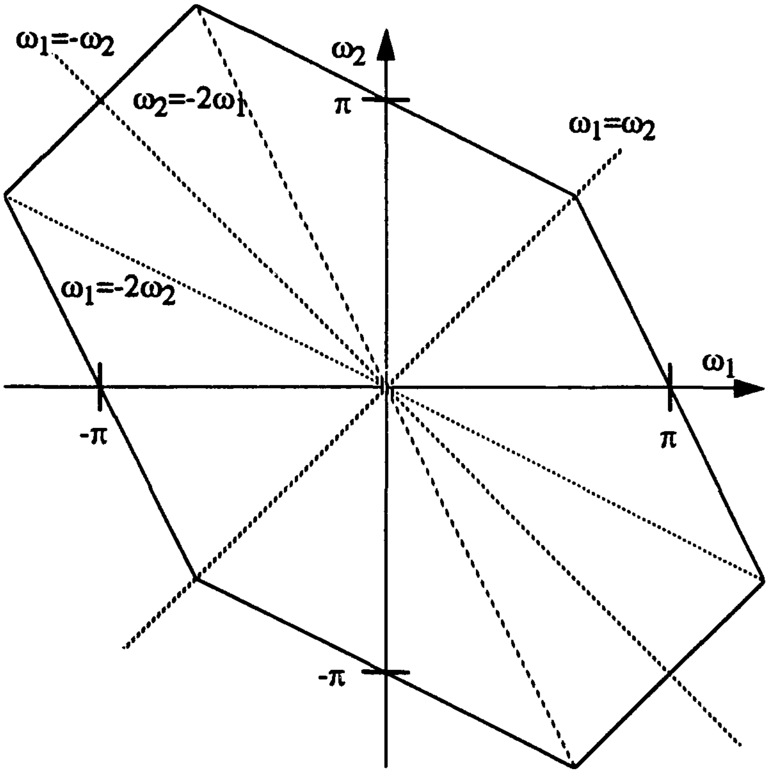


Figure 2.1: The bispectral plane.

Because of the symmetry described above, only a small portion of the bispectrum needs to be computed. Once the bispectrum is known for this small region known as the non-redundant region, the rest of the bispectrum can be found using symmetry. The non-redundant region, again for the sampled signal case, is shown in Figure 2.2. [Ref. 12,13]

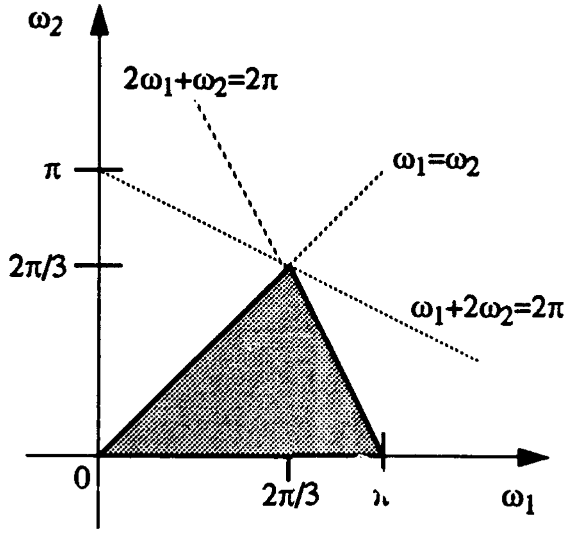


Figure 2.2: The non-redundant region of the bispectrum.

2. Computation by the Indirect Method

The indirect method requires a two-dimensional Fourier transform of an estimated third order cumulant sequence. A two-dimensional window can be used provided that it satisfies certain requirements based on cumulant symmetries. The following procedure outlines the indirect method given the data set $\{X(0), X(1), \ldots X(N-1)\}$: [Ref. 1]

1. Form K segments of M samples such that N=KM.
2. Remove the mean from each segment.
3. Defining i=1,2,...K to indicate the segment number, estimate the third order moment sequence for each segment, $\{x^{(i)}(k), k = 0, 1, \ldots M-1\}$:

$$\hat{r}^{(i)}(m,n) = \frac{1}{M}\sum_{l=s_1}^{s_2}[x^{(i)}(l)]^{*}x^{(i)}(l+m)x^{(i)}(l+n)\,; \tag{2.14}$$

where s1=max(0,-m,-n) and s2=min(M-1,M-1-m, M-1-n). This sequence is also called the tri-correlation sequence.

4. Average all segment moment estimates:

$$\hat{R}(m,n) = \frac{1}{K}\sum_{i=1}^{K}\hat{r}^{(i)}(m,n)\,. \tag{2.15}$$

5. Multiply the averaged moment estimate by a two-dimensional window, *W(m,n)*, and take the two-dimensional Fourier transform to obtain the bispectrum estimate:

$$\hat{B}(\omega_1,\omega_2) = \sum_{m=-L}^{L}\sum_{n=-L}^{L}\hat{R}(m,n)\,W(m,n)\,exp\{-j(\omega_1 m+\omega_2 n)\}\,; \tag{2.16}$$

where $L < M-1$.

The window, *W(m,n)*, used in the tri-correlation sequence defined in (2.16) must satisfy the following conditions [Ref. 1,11]:

1. The window conforms to third order cumulant symmetry. Specifically,

$$W(m,n) = W(n,m) = W(-m,n-m) = W(m-n,-n)\,.$$

2. The window is zero outside the region of support for the tri-correlaticn sequence.
3. The window is normalized to one at m=n=0.
4. The Fourier transform of the window is non-negative.

Windows used in this study belong to a separable class of window functions defined by

$$W(m,n) = d(m)\,d(n)\,d(n-m)\,, \tag{2.17}$$

where an appropriate one-dimensional window, *d(m)*, was chosen subject to the following constraints:

$$
\begin{aligned}
&d(m)=d(-m)\\
&d(m)=0 \quad \text{for } m > L,\\
&d(0)=1,\\
&D(\omega) \geq 0 \quad \text{for all } \omega. \quad \text{[Ref. 11]}
\end{aligned}
$$

The unit hexagonal window was constructed by applying (2.17) to a one-dimensional rectangular window [Ref. 14],

$$
\begin{aligned}
d(m) &= 1 \quad \text{for } |m| \leq L,\\
d(m) &= 0 \quad \text{otherwise.}
\end{aligned}
$$

The one-dimensional Parzen window, given by

$$
\begin{aligned}
d(m) &= 1 - 6\left(\frac{|m|}{L}\right)^2 + 6\left(\frac{|m|}{L}\right)^3 \quad \text{for } |m| \leq \frac{L}{2},\\
d(m) &= 2\left(1 - \frac{|m|}{L}\right)^3 \quad \text{for } \frac{L}{2} \leq |m| \leq L,\\
d(m) &= 0 \quad \text{for } |m| > L,
\end{aligned}
$$

was similarly transformed into a two-dimensional Parzen window [Ref. 1]. The last separable window considered is known as the minimum bias supremum window, or simply the optimum window. The corresponding one-dimensional window is defined by [Ref. 1]:

$$
\begin{aligned}
d(m) &= \frac{1}{\pi}\left|\sin\left(\frac{\pi m}{L}\right)\right| + \left(1 - \frac{|m|}{L}\right)\cos\left(\frac{\pi m}{L}\right) \quad \text{for } |m| \leq L,\\
d(m) &= 0 \quad \text{for } |m| > L.
\end{aligned}
$$

A trade-off situation exists between the Parzen window and the optimum window with regard to the windows' statistical properties. The bias of the optimum window is about 18% smaller than that of the Parzen window. However, the optimum window has a 26% greater variance. [Ref. 15]

3. Computation by the Direct Method

The direct method generates the bispectrum from (2.11). This requires a one-dimensional Fourier transform of the signal. Frequency domain smoothing can be used to reduce the variance of the final bispectrum estimate. The direct method is outlined for the data set $\{X(0), X(1), \ldots, X(N-1)\}$ [Ref. 1,12]:

1. The data sequence is segmented into K segments. The mean is removed from each segment before zero padding the segment to a convenient fast Fourier transform length, L.
2. An L-point Fourier transform, $Y^{(i)}(\lambda)$, is computed for each segment. The superscript designates the segment number.
3. The bispectrum estimate of the i^{th} segment, denoted as $\hat{b}_i$ is then computed by:
   $$\hat{b}_i(\lambda_1, \lambda_2) = \frac{1}{L} Y^{(i)}(\lambda_1) Y^{(i)}(\lambda_2) [Y^{(i)}(\lambda_1 + \lambda_2)]^* .$$
4. Two different approaches can be taken now to reduce the variance of the final estimate. The first approach, denoted $\hat{B}_1$, involves smoothing each segment separately and then averaging the smoothed segment estimates. The smoothing is accomplished uniformly over each bispectral location's R^2 neighboring frequency points [Ref. 12]:

   $$\hat{B}_1(\omega_1, \omega_2) = \frac{1}{K} \sum_{i=1}^{K} \frac{1}{R^2} \sum_{r=-R/2}^{R/2} \sum_{s=-R/2}^{R/2} \hat{b}_i(\lambda_1 + r, \lambda_2 + s) \qquad (2.18)$$

   where $\omega_1 = (\frac{2\pi f_s}{L}) \lambda_1$, $\omega_2 = (\frac{2\pi f_s}{L}) \lambda_2$, and $R \leq L$.

   In the second approach, denoted $\hat{B}_2$, all the unsmoothed segment estimates are first coherently averaged:

   $$\hat{B}_2(\omega_1, \omega_2) = \frac{1}{K} \sum_{i=1}^{K} \hat{b}_i(\omega_1, \omega_2) . \qquad (2.19)$$

Then a frequency domain smoothing window is applied through a two-dimensional convolution. [Ref. 14]

Hi-Spec software [Ref. 14] is used in this study to compute the bispectrum by the direct method. This program uses the second approach described above, based upon (2.19) to reduce the variance of the final bispectrum estimate. There are a few options available in this software package to accomplish the frequency domain smoothing. The option chosen most frequently is the Rao-Gabr window defined by:

$$W(m,n) = \frac{\sqrt{3}}{\pi^3}\left[1 - \frac{m^2+n^2+mn}{M^2}\right] \quad \text{for } (m,n) \in G;$$

where $M = \frac{L}{2}$, or half the Fourier transform length. The set G is the collection of (m,n) points satisfying

$$m^2+n^2+mn \le \frac{winlen}{M^2}.$$

The parameter *winlen* is the desired window length. Spatial smoothing is applied over the $(winlen)^2$ adjacent frequency points around each bispectrum point. [Refs. 12,14]

4. Comparison of Methods

Both the direct method and the indirect method lead to asymptotically unbiased estimators, i.e., $E\{\hat{B}(\omega_1,\omega_2)\} \cong B(\omega_1,\omega_2)$. The indirect method has an asymptotic variance given by:

$$\begin{aligned}\text{var}\{\text{Re }\hat{B}(\omega_1,\omega_2)\} &= \text{var}\{\text{Im }\hat{B}(\omega_1,\omega_2)\} \\ &= \frac{\varepsilon_w}{M^2K}P(\omega_1)P(\omega_2)P(\omega_1+\omega_2);\end{aligned}$$

where ε_w is the energy of the tri-correlation domain window, $P(\omega)$ is the true power spectrum, $M = 2L+1$ is the Fourier transform size (i.e., an $M \times M$ FFT), and K is the number of segments averaged [Ref. 1]. The asymptotic variance of the direct method when using the smoothing approach described by (2.18) is:

$$\begin{aligned}\operatorname{var}\{\operatorname{Re}\hat{B}(\omega_1,\omega_2)\} &= \operatorname{var}\{\operatorname{Im}\hat{B}(\omega_1,\omega_2)\}\\ &= \frac{N}{2KL^2}P(\omega_1)P(\omega_2)P(\omega_1+\omega_2);\end{aligned}$$

where N is the length of the data sequence, K is the number of segments, and L is the size of each segment. [Ref. 12]

The direct method and the indirect method are identical when neither uses a window, and the indirect method's tri-correlation estimate is computed for a number of lags equal to the full segment length so that $L = M - 1$ in (2.16).

C. THE 1-1/2 D INSTANTANEOUS POWER SPECTRUM

The 1-1/2 D instantaneous power spectrum (1-1/2 D_{ips}) is a combination of the standard 1-1/2 D spectrum (1-1/2 D_{std}) and the Instantaneous Power Spectrum (IPS) [Refs. 3,5,16]. It is shown in [Ref. 5] that 1-1/2 D_{ips} performs better than the conventional spectrogram in some respects. When used to process dynamic signals, 1-1/2 D_{ips} is observed to have a quicker spectral rise time and a quicker fall-off time than the spectrogram. The 1-1/2 D_{ips} method also appears to be good at detecting low SNR stationary signals. [Ref. 5]

The standard 1-1/2 D spectrum is a degenerate form of the bispectrum. When the first lag in (2.7) to set to zero, the third order cumulant expression becomes

$$\begin{aligned}C^{(3)}(0,l_2) &= E\{X^*(n)X(n)X(n+l_2)\}\\ &= E\{|X(n)|^2X(n+l_2)\}.\end{aligned} \tag{2.20}$$

A one-dimensional Fourier transform of (2.20) produces the 1-1/2 D_{std} spectrum. [Ref. 3]

The Instantaneous Power Spectrum is defined as the average of the derivatives of Page's running spectrum,

$$\rho(t,f) = \frac{\partial}{\partial t}\left|\int_{-\infty}^{t} s(\tau)\,exp(-j2\pi f\tau)\,d\tau\right|^2 , \tag{2.21}$$

and Levin's backward running spectrum,

$$\rho^b(t,f) = \frac{\partial}{\partial t}\left|\int_{t}^{\infty} s(\tau)\,exp(-j2\pi f\tau)\,d\tau\right|^2 . \tag{2.22}$$

A discrete version of IPS is obtained by taking the Fourier transform of windowed correlation estimates formed using simple lag products vice a sum of lagged products:

$$IPS(n,\omega) = \frac{1}{2}\sum_{k=0}^{N-1}\{X(n)X^*(n-k) + X^*(n)X(n+k)\}\,w(k)\,exp(-j\omega k)\,; \tag{2.23}$$

where $w(k)$ is the window function, and N is the length of the sampled data sequence $\{X(0), X(1), \ldots, X(N-1)\}$ [Refs. 5,16].

The 1-1/2 D_{ips} algorithm is created by substituting the estimate of (2.20) into (2.23) so that

$$1\text{-}1/2\ D_{ips}(n,\omega) = \frac{1}{2}\sum_{k=0}^{N-1}\{|X(n)|^2X^*(n-k) + |X(n)|^2X(n+k)\}\,w(k)\,exp(-j\omega k) \tag{2.24}$$

with variables defined as in the IPS equation. [Ref. 5]

D. INSTANTANEOUS HIGHER-ORDER MOMENT SLICE

The time-frequency representation introduced in [Ref. 6] is based upon a one-dimensional Fourier transform of a higher-order moment slice. While satisfying certain constraints, a desired number of different moment slice sets are generated in order to form an equal number of different but related time-frequency surfaces. As a consequence of the moment slice construction constraints, signal dependent auto-terms are enhanced and c.:oss-terms are de-emphasized when the time-frequency surfaces are coherently summed. To further control the contrast between auto-terms and cross-terms, each moment slice is multiplied by a kernel function before being Fourier transformed. The kernel function controls the degree of auto-term and cross-term concentration on the time-frequency surface. [Ref. 6]

The instantaneous higher-order moment slice (IHOMS) method is used to detect multicomponent linear chirp signals having different instantaneous frequency slopes, or slew rates. One characteristic of this time-frequency representation is that auto-term lines pass through the origin of the time-frequency plane. The number of auto-terms so identified indicate the number of chirps in the signal. In addition, the angle that the auto-term line makes with the time axis is related to the instantaneous frequency of the chirp. [Ref. 6]

The IHOMS method is outlined for a received signal, $x(t)$, given by

$$x(t) = \sum_{i=1}^{M} exp\,[j(u_i t + b_i t^2)]\,, \tag{2.25}$$

where M is the number of chirps in the received signal. The instantaneous frequency of the ith chirp is given by:

$$\begin{aligned} f_i(t) &= \frac{1}{2\pi}\frac{d}{dt}[u_i t + b_i t^2] \\ &= \frac{1}{2\pi}[u_i + 2b_i t]\,. \end{aligned} \tag{2.26}$$

A general expression for a sampled version of (2.25) is:

$$x(n) = \sum_{i=1}^{M} exp\,\{j2\pi\left[\,(\frac{n}{N})\,u_i + (\frac{n}{N})^2\,b_i\right]\}\,; \tag{2.27}$$

where the sampling period is assumed to be unity, and N is the number of discrete samples in the data record. [Ref. 6]

The following procedure is used to form the IHOMS time-frequency surface [Ref. 6]:

1. The desired order of the moment slice, r, is chosen first. The order should be greater than three so that a variety of related time-frequency surfaces can be summed in order to form the composite surface. In general, auto-terms stand out better when more surfaces are summed. However, if the order is chosen too large, cross-term effects increase and may cause spurious peaks in the composite time-frequency surface.
2. The k^{th} parameter vector, $\underline{a}^{(k)} = \left[1 \; a_1^{(k)} \; \dots a_{r-1}^{(k)}\right]$, is generated somewhat arbitrarily but subject to the following constraints:

$$\sum_{i=1}^{r-1} a_i^{(k)} = 1, \tag{2.28}$$

and

$$\sum_{i=1}^{r-1} [a_i^{(k)}]^2 = 1. \tag{2.29}$$

3. The moment slice is then calculated for a given instant of time (t), a given parameter vector, and a desired number of lags (τ):

$$m_r^x(\tau;t,\underline{a}^{(k)}) = x^*(\tau)\,x(t+a_1\tau)\dots x(t+a_{r-1}\tau)\,. \tag{2.30}$$

4. The parameter g in the kernel function, $exp(-g\tau^2)$, is chosen very small so that auto-terms concentrate along straight lines. However, if g is made too small, cross-terms show up everywhere on the time-frequency plane and tend to add up like auto-terms when the surfaces are later summed to form the composite surface.

5. The Fourier transform of the moment slice is computed and multiplied by the kernel function,

$$Z(t,f;\underline{a}^{(k)}) = \int_{-\infty}^{\infty} exp(-g\tau^2)\, m_r^x(\tau;t,\underline{a}^{(k)})\, exp(-j2\pi f\tau)\, d\tau. \tag{2.31}$$

6. Steps three through five are repeated for a different time instant, the same parameter vector, the same g, and the same number of lags. Stacking the results of (2.31) for successive time instants forms a complex valued time-frequency surface based upon a particular parameter vector.

7. A new parameter vector is generated and then steps three through six are repeated to create another time-frequency surface. This is done again and again until the desired number of complex valued surfaces are computed. The composite surface is obtained by taking the magnitude of the coherently summed parameter vector dependent time-frequency surfaces,

$$G(t,f) = \left|\sum_{k=1}^{K} Z(t,f;\underline{a}^{(k)})\right|, \tag{2.32}$$

where K is the number of different parameter vectors used to generate an equal number of related time-frequency surfaces.

The maxima of the function,

$$D(\theta) = \sum_{v=R_1}^{R_2} G(v\cos\theta, v\sin\theta), \tag{2.33}$$

where θ is varied between zero and $\pi/2$ radians, and the radial distance v is varied incrementally between R_1 and R_2, correspond to chirps. The value of b in the slope term of a given chirp's instantaneous frequency expression, (2.26), can be found from the θ producing the associated maximum value in (2.33) using the relation:

$$b_j = \frac{N^2}{2\beta}\tan\theta_j; \tag{2.34}$$

where the subscript j denotes a particular chirp, β is the number of lags used to compute the moment slice in (2.30), and N is defined as in (2.27).

III. PROCESSING GAIN COMPARISONS

In this chapter we compare the performance of 1-1/2 D_{ips} to that of the periodogram and its time-frequency representation, the spectrogram. The next chapter will examine bispectral as well as IHOMS methods, and compare them to 1-1/2 D_{ips} and the spectrogram where appropriate.

Matlab programs used to conduct the computer simulations for this chapter are described first. Theoretical calculations and simulation results are then presented to show how each method treats noise and signal components separately. Finally, relative signal-to-noise ratio processing gains are estimated.

A. SIMULATION PROGRAMS

The 1-1/2 D_{ips} algorithm was originally implemented in [Ref. 5] with the extrinsic Matlab function ONE_HALF. Modifications were subsequently made to the program to remove the mean from each data segment and to delete some unnecessary smoothing steps. The revised ONE_HALF function is contained in Appendix C. The user specifies an input data sequence, the size of the data window, a step increment, and a window function. ONE_HALF returns the time-frequency representation of the input sequence. The ONE_HALF algorithm is based upon (2.24) but is constructed in such a way as to avoid undesirable product term interferences that may form spurious peaks in the time-frequency representation. The following sequence is first generated:

$$prod1 = \left[element(0)\ element(1)\ ...element(\frac{winlen}{2} - 1) \right]; \tag{3.1}$$

where,

$$element(k) = |x(n)|^2 \{x^*(n-k) + x(n+k)\}, \tag{3.2}$$

winlen is the specified data window length, and *x(n)* is the input data segment. A second sequence, *prod2*, is formed by deleting the first element in *prod*1 and then reversing the order of the conjugated remaining elements:

$$prod2 = \left[prod1^*\left(\frac{winlen}{2}\right), prod1^*\left(\frac{winlen}{2} - 1\right), \ldots, prod1^*(2)\right]. \tag{3.3}$$

The two sequences are then concatenated and a zero is inserted between them to form a proper correlation function having a real valued Fourier transform. The 1-1/2 D_{ips} estimate can then be obtained by multiplying the real part of the transformed sequence by one-half. However, since the magnitude squared estimate is required later in this study to compare the 1-1/2 D_{ips} and spectrogram representations, ONE_HALF computes the magnitude vice the real part of the transform:

$$1\text{-}1/2\ D_{ips} = \left|\frac{1}{2}\Im\{[prod1\ 0\ prod2]\}\right|; \tag{3.4}$$

where $\Im$ denotes Fourier transform.

SPECTRO is the extrinsic Matlab function included in Appendix C that implements the spectrogram. It has the same input and output format as ONE_HALF. The two functions are equivalent with respect to zero padding and transform length so that fair comparisons can be made between the two methods. At each step the function computes the periodogram:

$$P_{per}(\omega) = \left|\sum_{n=0}^{winlen-1} x(n)\,w(n)\,exp(-j\omega n)\right|^2; \tag{3.5}$$

where the data segment, *x(n)*, is padded with zeros to the transform length of *winlen* if necessary, and *w*(*n*) is an appropriate window function. A rectangular window (i.e., *w*(*n*) = 1 over the support of *n*) is assumed in the estimate variance calculations to follow.

B. NOISE ONLY PERFORMANCE COMPARISON

The variance of the 1-1/2 D_{ips} estimate is derived for a zero mean white Gaussian input. The details of this calculation are contained in Appendix A. If the input is real, the theoretical variance based on (2.24) is given by,

$$\text{var}\ \{1\text{-}1/2\ \hat{D}_{ips}\} = \frac{54 + 6winlen}{4}\,(\sigma_x^2)^3; \qquad (3.6)$$

where *winlen* is the length of the data window and σ_x^2 is the variance of the input. For a complex input the variance is

$$\text{var}\ \{1\text{-}1/2\ \hat{D}_{ips}\} = [winlen + 2]\,(\sigma_x^2)^3. \qquad (3.7)$$

The variance of the periodogram as defined by (3.5) with a rectangular window is [Refs. 2,11]:

$$\text{var}\ \{Per\} \cong winlen^2\,(\sigma_x^2)^2. \qquad (3.8)$$

Both estimate variances are dependent upon the length of the data window and the input variance. For a given input variance and increasing window length, the periodogram's variance increases more rapidly than does the estimate variance of 1-1/2 D_{ips} because the periodogram's variance is proportional *winlen*^{2} rather than *winlen.* However, since the 1-1/2 D_{ips} variance is a function of the input variance cubed and the periodogram's variance depends upon input variance squared, 1-1/2 D_{ips}'s variance increases quicker than the periodogram's for a fixed window length and increasing input variance.

Computer simulations confirm the theoretical variance expressions (3.6) and (3.8), and the expected trends as window length and input variance are changed. Real valued white Gaussian noise is processed by ONE_HALF employing a rectangular window and a step increment set equal to the window length so that no overlapping of data segments occurs. Since SPECTRO is written for a fixed step increment of one, the periodogram is estimated directly from (3.5) in the Matlab script file used to perform the simulations. Simulation parameters, measured estimate variances, and theoretically expected variances are shown in Table 3.1. Measured input variances are used to compute the theoretical variances. There is good agreement between theoretical and expected values. The methods cannot be compared number for number to each other because the periodogram variance is a magnitude squared quantity and 1-1/2 D_{ips} is a magnitude quantity. However, comparisons between the methods can be made regarding their relative responses to changing window length and input variance. For a window length of 256, 1-1/2 D_{ips}'s variance increases by a factor of about 342 when input variance is increased from one to seven. For the same variance change, the periodogram's variance increases by a considerably smaller factor of about 49. When input variance is three and window length increases from 128 to 512 the periodogram's variance increases by a factor of 15 while 1-1/2 D_{ips} only increases by a factor of three.

TABLE 3.1: THEORETICAL VS. MEASURED 1-1/2 D_{IPS} AND PERIODOGRAM VARIANCES FOR VARIOUS WINDOW LENGTHS AND INPUT VARIANCES

input variance	window length	1-1/2 D_{ips} theoretical variance	1-1/2 D_{ips} measured variance	periodogram theoretical variance	periodogram measured variance
1	128	204.3	205.5	20,456	19,487
1	256	394.2	377.8	77,549	93,217
1	512	779.8	707.6	303,490	297,930
3	128	5,515.8	5,549.5	184,100	175,380
3	256	10,644	10,201	697,940	838,960
3	512	21,056	19,107	2,731,400	2,681,300
7	128	70,071	70,499	1,002,300	954,850
7	256	135,220	129,590	3,799,900	4,567,600
7	512	267,490	242,720	14,871,000	14,598,000

The 1-1/2 D_{ips} and spectrogram time-frequency representations of a white Gaussian noise input are shown in Figure 3.1. ONE_HALF generates the 1-1/2 D_{ips} estimate and SPECTRO produces the spectrogram. Both representations are formed by applying a rectangular window to a single 128 noise sample realization using a 64 point window length and a step increment of one. The most striking difference between the two surfaces is that 1-1/2 D_{ips} consists of several short time duration spikes while the spectrogram exhibits a more continuous and smoother surface. The spikier appearance of the 1-1/2 D_{ips} surface indicates that the 1-1/2 D_{ips} estimate de-correlates faster than the spectrogram. As a consequence, the typical 1-1/2 D_{ips} frequency bin contains more independent segments than does the typical spectrogram bin. This is significant because a larger reduction in variance is realizable from a greater number of independent segments when the power in a bin is summed.

The number of independent segments in a bin is estimated by applying a relation used to determine the degrees of freedom for a chi-squared distribution. Given a collection of *n* normally distributed zero mean unit variance random variables, the degrees of freedom (DF) represent the number of independent random variables contained in the collection. Degrees of freedom are found from the mean and variance of the sum of the random variables, $s = x_1 + x_2 + \ldots + x_n$, using the relation:

$$DF = \frac{2\mu_s^2}{\sigma_s^2}; \tag{3.9}$$

where μ_s and σ_s^2 are the mean and variance respectively, of s. [Ref. 7]

Ten spectrogram and 1-1/2 D_{ips} representations for different realizations of real white unit variance Gaussian noise are used to estimate the degrees of freedom with (3.9). The degrees of freedom are calculated for the individual time-frequency cell (i.e., the estimate's

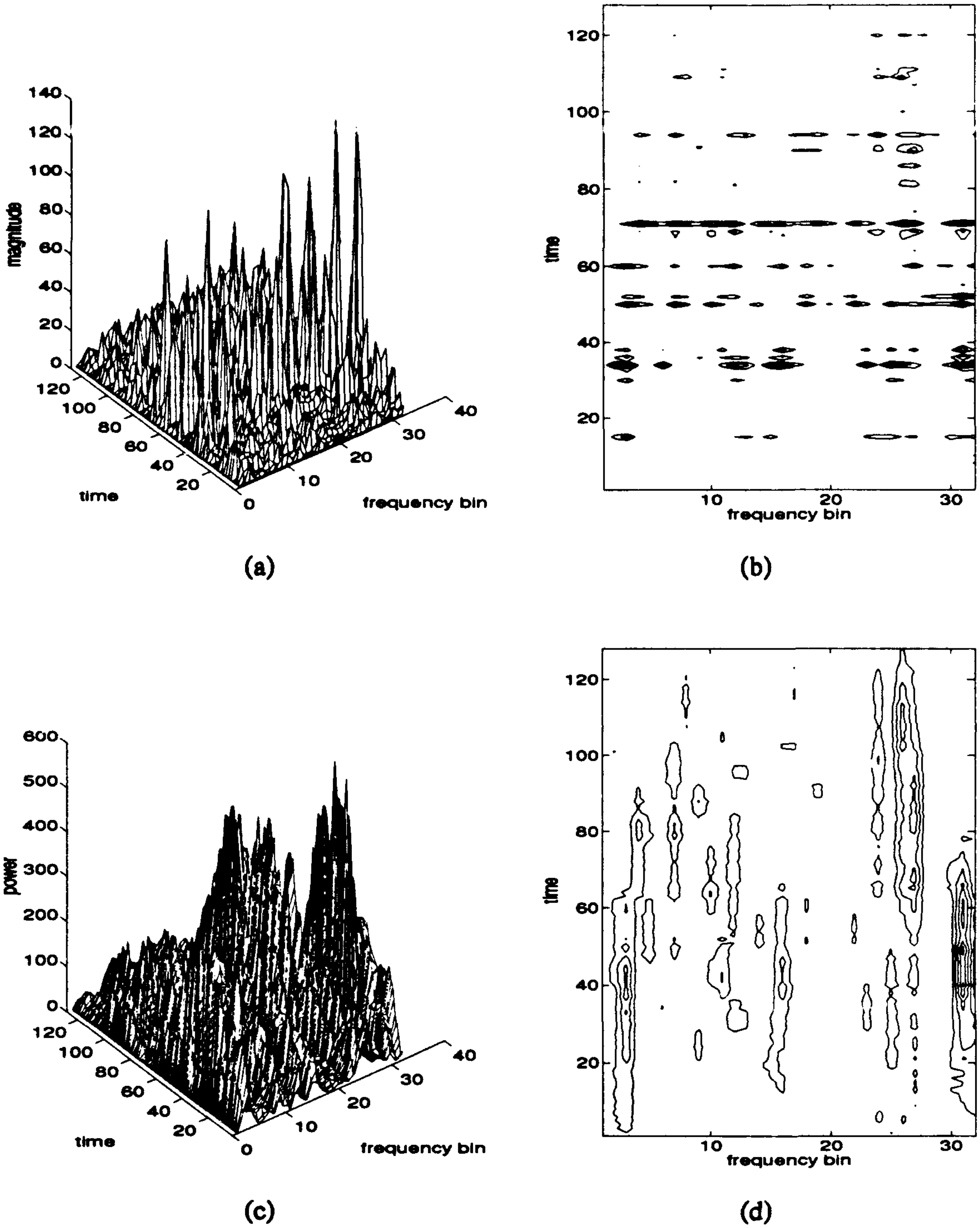


Figure 3.1: Time-frequency representations of unit variance white Gaussian noise; (a) 1-1/2 D_{ips} mesh plot, (b) 1-1/2 D_{ips} contour plot, (c) spectrogram mesh plot, and (d) spectrogram contour plot.

value in one frequency bin at one time step), as well as for the frequency bin. The degrees of freedom for a cell is based upon the mean and variance of all the values in the frequency bin. The degrees of freedom for a bin is based upon the mean and variance of the average frequency bin values for all the bins in the surface. The final degrees of freedom estimates are obtained by averaging the degrees of freedom computed for each of the ten realizations. Table 3.2 summarizes the results of the simulations for various window lengths and window functions. In all cases, SPECTRO and ONE_HALF use a step increment of one, and the data sequence is twice the size of the window length. The degrees of freedom for a cell in the 1-1/2 D_{ips} representation is approximately 0.6 while the degrees of freedom for the spectrogram cell is about four. The degrees of freedom for a bin depends upon the window function. With a rectangular window, a 1-1/2 D_{ips} bin has about 17.5 degrees of freedom and a spectrogram bin has almost six degrees of freedom. If a Hamming window is applied to the data, the spectrogram and 1-1/2 D_{ips} bin degrees of freedom figures increase to about eight and 27, respectively.

TABLE 3.2: DEGREES OF FREEDOM FOR 1-1/2 D_{IPS} AND SPECTROGRAM ESTIMATES

winlen	window function	ONEHALF DF_{cell}	ONEHALF DF_{bin}	SPECTRO DF_{cell}	SPECTRO DF_{bin}
64	rectangular	0.6582	16.3737	4.1109	5.8239
64	Hamming	0.6185	26.9717	4.4057	7.7394
128	rectangular	0.6326	18.7693	4.0822	5.9613
128	Hamming	0.5974	27.5028	4.1742	8.0081

The data sequence length divided by the bin degrees of freedom indicate the size of the independent segments in a typical frequency bin. For example, the size of an independent segment in the 1-1/2 D_{ips} estimate based on a window length of 64, a 128 point sequence, and a rectangular window is $128/16.3737 \cong 8$. For the same parameters, the spectrogram independent bin segment size is about 22. These figures seem reasonable based upon "eyeball" estimates of independent bin segment size in Figure 3.1 contour plots.

When an estimate's frequency bin values are averaged, the anticipated variance reduction factor is given by the ratio of that estimate's DF_{bin} to DF_{cell} values. The expected variance reduction factor for 1-1/2 D_{ips} is about 29 for a rectangular window and 45 for a Hamming window. The spectrogram has an expected variance reduction factor of 1.5 for a rectangular window and two for a Hamming window.

Computer simulations using ONE_HALF and SPECTRO confirm that an exploitable difference in variance reduction capability exists between 1-1/2 D_{ips} and the spectrogram when frequency bin averages of a time-frequency surface are computed. Gaussian white noise is simulated to serve as a noise only, real input signal. Both methods employ a rectangular window on the same input signal. The step increment is specified to be one for each method and the data window length is consistently set equal to half of the data sequence length. For the first set of simulations, noise variance is fixed while the transform length is varied. The simulations require that each method generate a time-frequency representation for fifteen different input sequences. The fifteen representations are then averaged to form one time-frequency surface. For this, and all remaining simulations performed in this study, the averaged 1-1/2 D_{ips} surface is squared so that its amplitude can be fairly compared to the spectrogram's amplitude. As a consequence of computer program design, data sequence length, and choice of step size the first and last *winlen/2* time steps are calculated using zero-padded data segments. These time steps are deleted and the average value of each frequency bin on the average surface is calculated.

Figures 3.2 through 3.6 show these average frequency bin values for an input noise variance of one and window lengths of 32, 64, 128, 256, and 512, respectively. The bin corresponding to dc, and the bin corresponding to the fold-over frequency are omitted from the plots. To allow for a meaningful comparison, the mean frequency bin average is subtracted from each plot. Examination of Figures 3.2 through 3.6 indicate that 1-1/2 D_{ips} exhibits a lesser output variation than does the spectrogram as the transform length increases. This observation is confirmed in Figure 3.7 which plots the measured output variance versus transform length for the first simulation set.

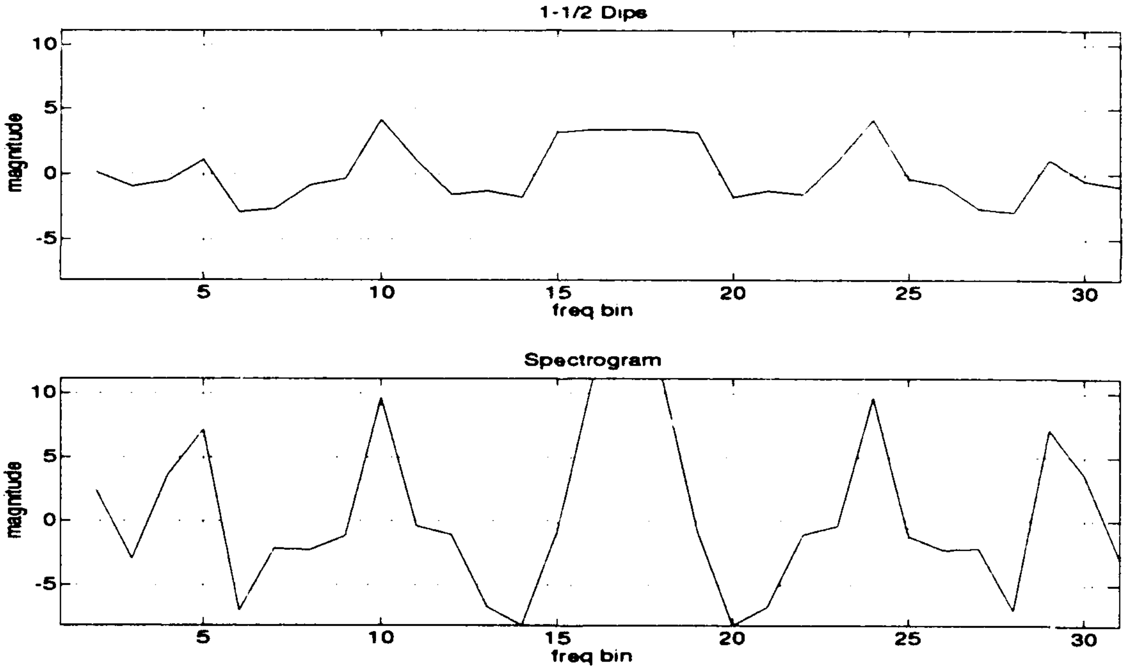


Figure 3.2: Averaged output variance for *winlen* = 32 and $\sigma_{in}^2 = 1$.

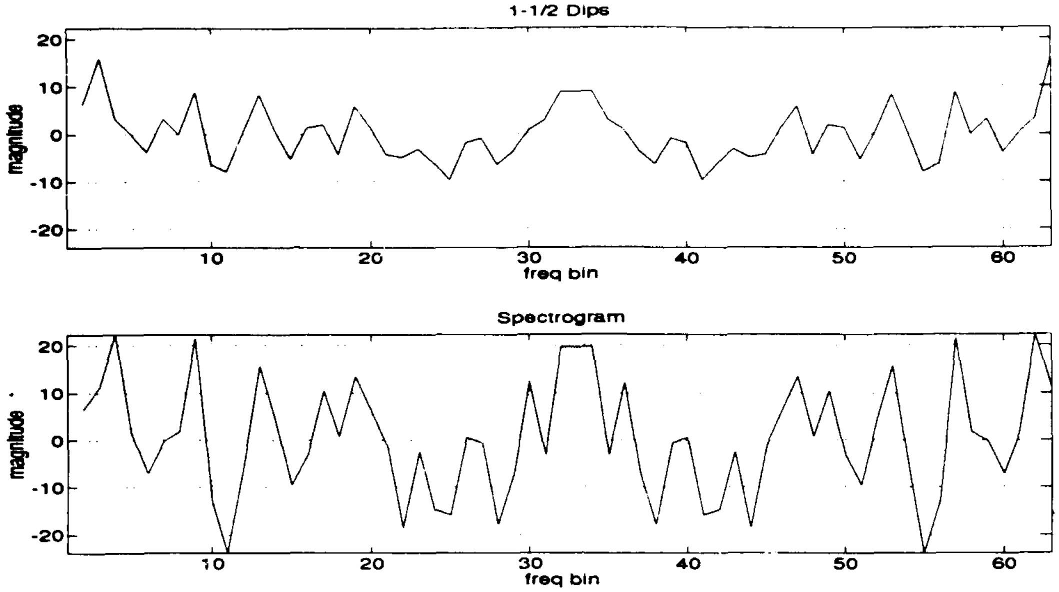


Figure 3.3: Averaged output variance for *winlen* = 64 and $\sigma_{in}^2 = 1$.

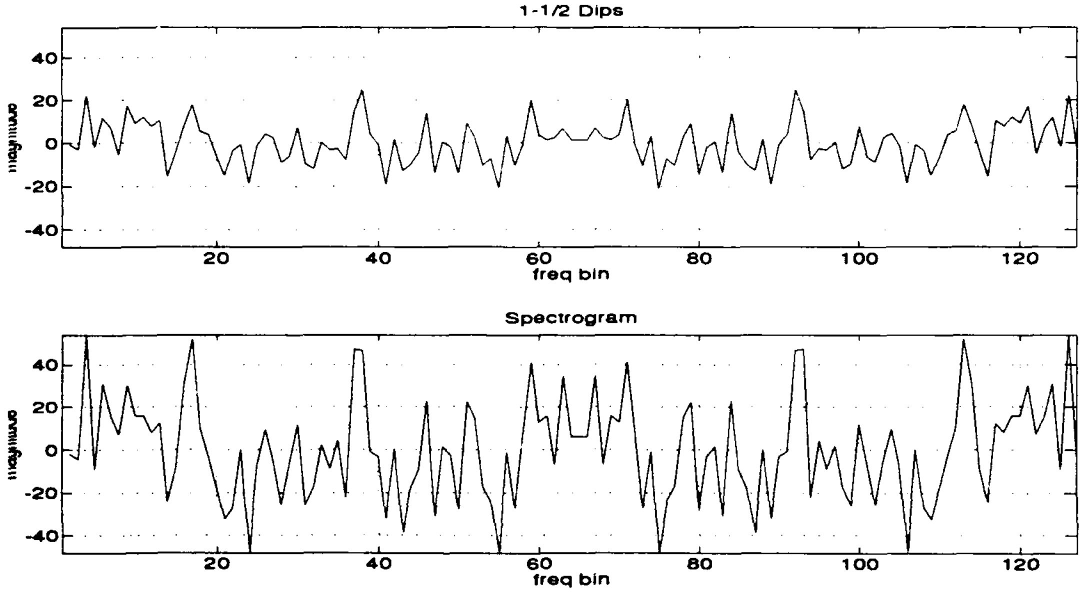


Figure 3.4: Averaged output variance for *winlen* = 128 and $\sigma_{in}^2 = 1$.

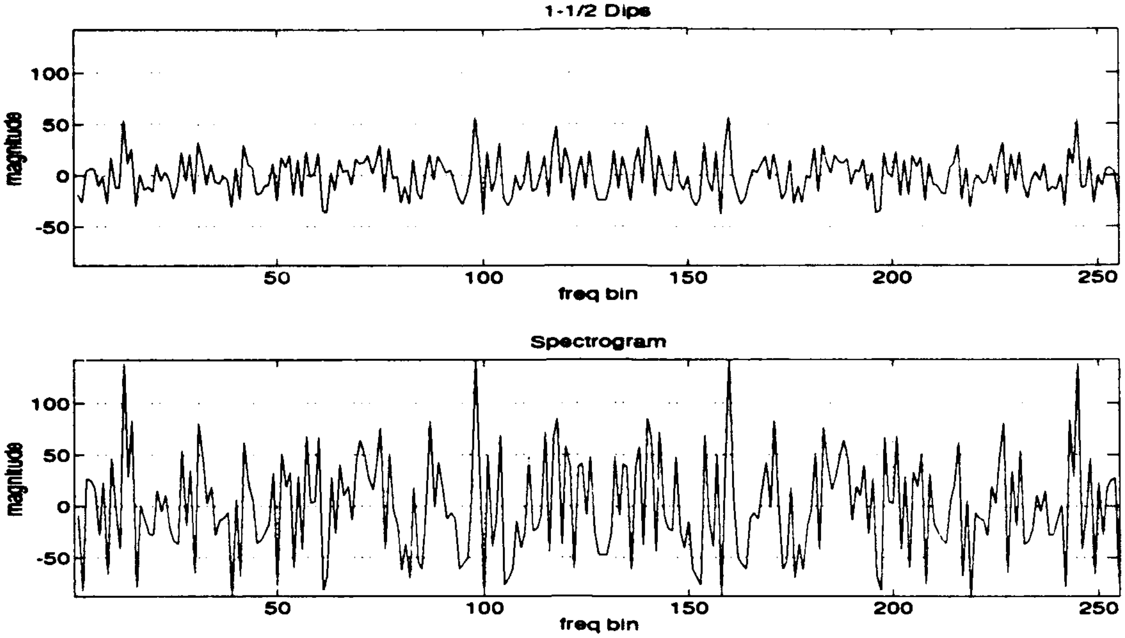


Figure 3.5: Averaged output variance for *winlen* = 256 and $\sigma_{in}^2 = 1$.

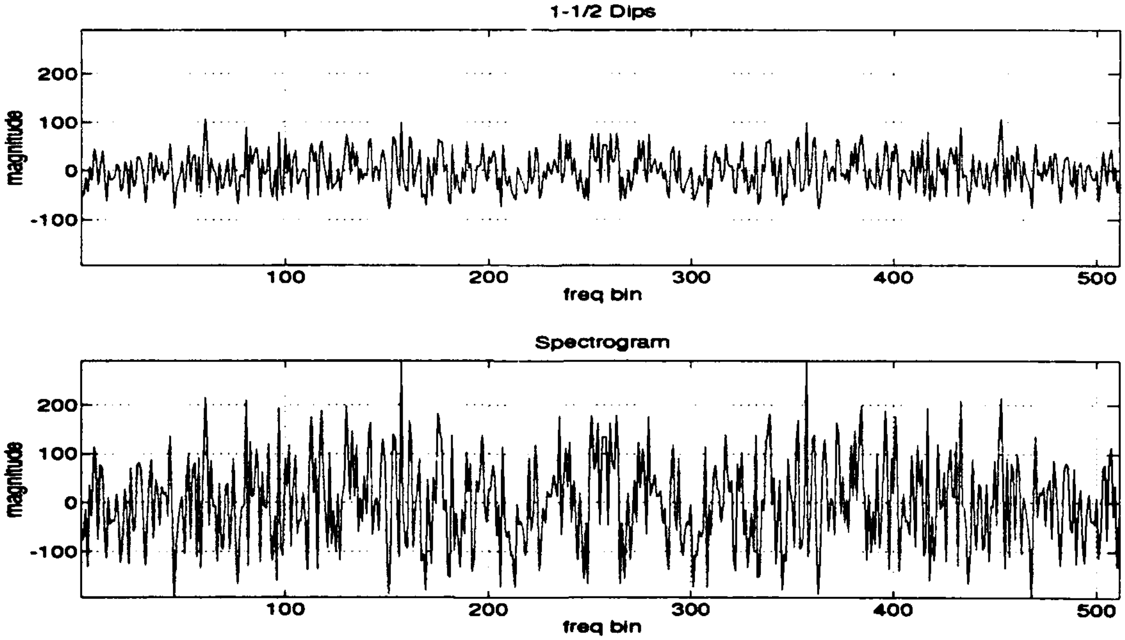


Figure 3.6: Averaged output variance for *winlen* = 512 and $\sigma_{in}^2 = 1$.

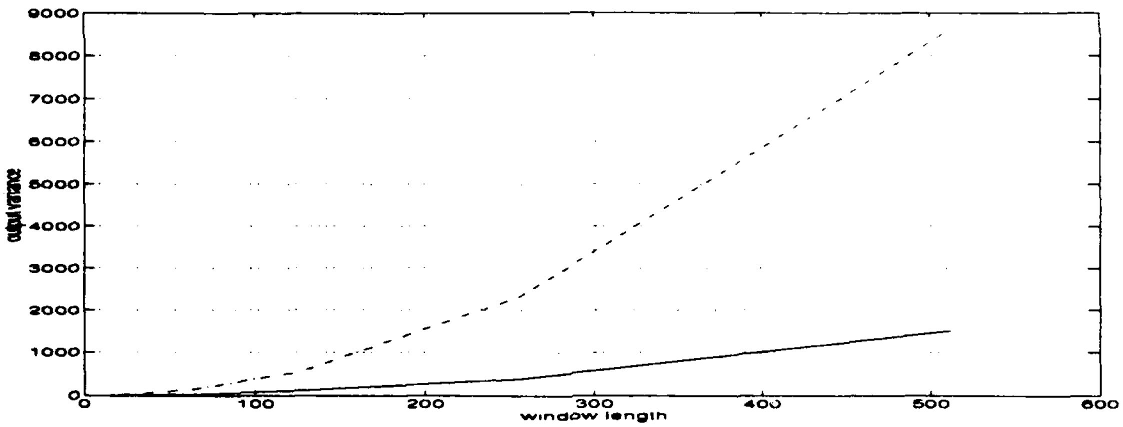


Figure 3.7: Spectrogram (dashed) and 1-1/2 D_{ips} (solid), output variance vs. window length for $\sigma_{in}^2 = 1$.

The simulation was repeated for different input noise variances and a fixed window length. Figure 3.8 shows the relationship between output variance and input variance for a window length of 128. The point at which the two curves in Figure 3.8 intersect can be considered a performance breakpoint. At input variances below the breakpoint value (i.e., about 1.5 from Figure 3.8), the 1-1/2 D_{ips} estimate has less variation than the spectrogram. When the input variance is larger than the breakpoint value the spectrogram exhibits less variation than 1-1/2 D_{ips}. Other simulations show that breakpoints for window lengths ranging from 32 to 512, although gradually increasing as window length increases, all correspond to an input noise variance of approximately 1.5.

Figure 3.8 shows that the spectrogram has a variance of about 1000 when the input variance is about 1.3. Considering the processing scheme, the expected variance is the theoretical estimate variance given by (3.8) reduced by two factors. The first factor is equal to the number of time-frequency surfaces averaged to obtain a representative surface. The second factor is the degrees of freedom based variance reduction factor estimated earlier. Therefore, the expected spectrogram output variance is given by:

$$\begin{aligned}\text{var }\{Spectro\} &\cong \frac{winlen^2\,(\sigma_x^2)^2}{nsurf\cdot VRF}\\ &= \frac{128^2\cdot 1.3^2}{15\cdot 1.5}\\ &= 1231;\end{aligned}$$

where the numerator is (3.8), *nsurf* is the number of time-frequency surfaces averaged in the simulation, and *VRF* is the degrees of freedom variance reduction factor for a rectangular window. The expected and actual variances for the spectrogram agree fairly well considering that the comparison is based upon approximate values.

The same comparison is made for 1-1/2 D_{ips}. According to Figure 3.8, an input variance of approximately 1.45 results in a 1-1/2 D_{ips} estimate variance of about 1000. The theoretical variance given by (3.6) is reduced by the same factor of 15 used in the spectrogram calculation since this factor depends only upon the number of time-frequency surfaces averaged. The theoretical variance is also reduced by a degrees of freedom variance reduction factor that is larger than the spectrogram's (29 vice 1.5). To account for squaring the output of ONE_HALF in the simulations, (3.6) is also squared. The expected 1-1/2 D_{ips} variance based on the processing scheme is:

$$\begin{aligned}\text{var }\{1\text{-}1/2\ \hat{D}_{ips}\} &\cong \frac{[54+6winlen]^2\,(\sigma_x^2)^6}{4^2\cdot nsurf\cdot VRF}\\ &= \frac{[54+6\cdot 128]^2\,(1.45)^6}{4^2\cdot 15\cdot 29}\\ &= 902.\end{aligned}$$

The expected variance agrees fairly well with the simulation variance for the 1-1/2 D_{ips} estimate considering that approximate values are being compared.

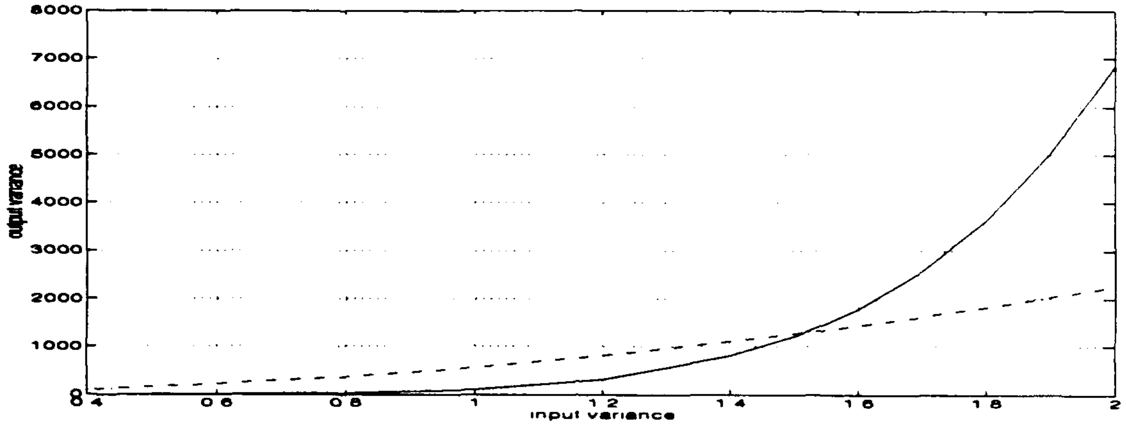


Figure 3.8: Spectrogram (dashed) and 1-1/2 D_{ips} (solid), output variance vs. input variance for *winlen* = 128.

C. SIGNAL ONLY PERFORMANCE COMPARISON

Appendix B details the calculations made using (2.24) to determine the theoretical 1-1/2 D_{ips} estimate for a signal only. For a real valued sinusoid given by

$$x(n) = \cos\left(2\pi\frac{m}{N}n\right), \tag{3.10}$$

the theoretical estimate is:

$$\text{1-1/2 } D_{ips}(n, \omega) = \frac{N}{2}\cos^3\left(2\pi\frac{m}{N}n\right)\{\delta(\omega - m) + \delta[\omega - (N - m)]\}. \tag{3.11}$$

The periodogram is defined in (3.5). For the same signal, the theoretical value of the periodogram is the magnitude squared of the Fourier transform of the input signal:

$$Per(n, \omega) = \left(\frac{N}{2}\right)^2\{\delta(\omega - m) + \delta[\omega - (N - m)]\} \tag{3.12}$$

If the signal consists of a single complex sinusoid,

$$x(n) = exp\left(j2\pi\frac{m}{N}n\right), \tag{3.13}$$

the theoretical 1-1/2 D_{ips} estimate is given by:

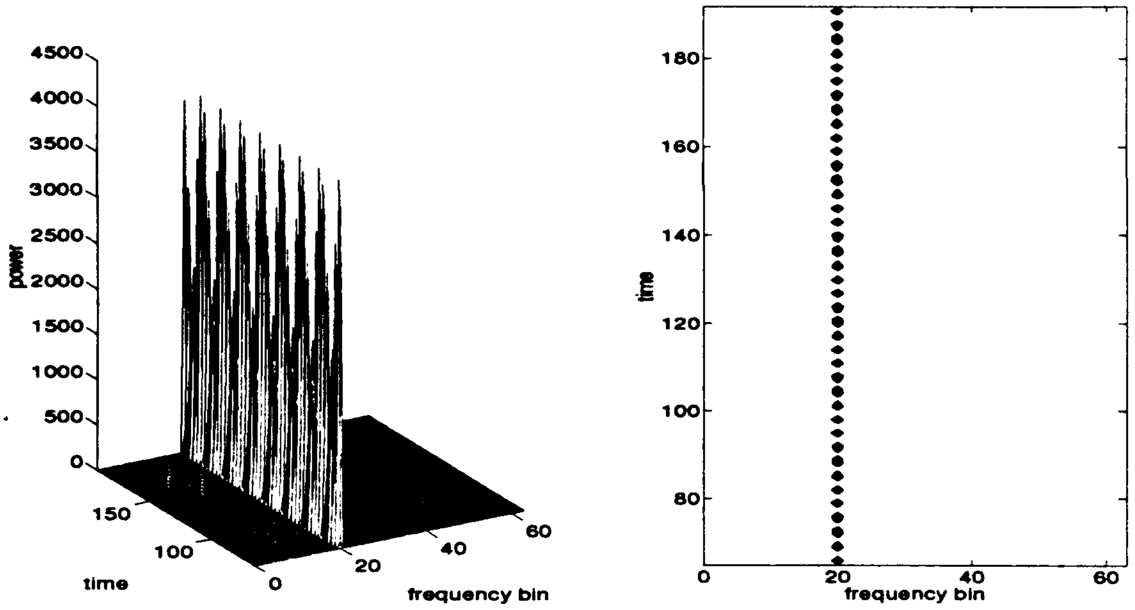
$$1\text{-}1/2\ D_{ips}(n,\omega) = N\cos\left(2\pi\frac{m}{N}n\right)\delta(\omega-m)\,; \tag{3.14}$$

and the theoretical value of the periodogram is:

$$Per(n,\omega) = N^2\delta(\omega-m) \tag{3.15}$$

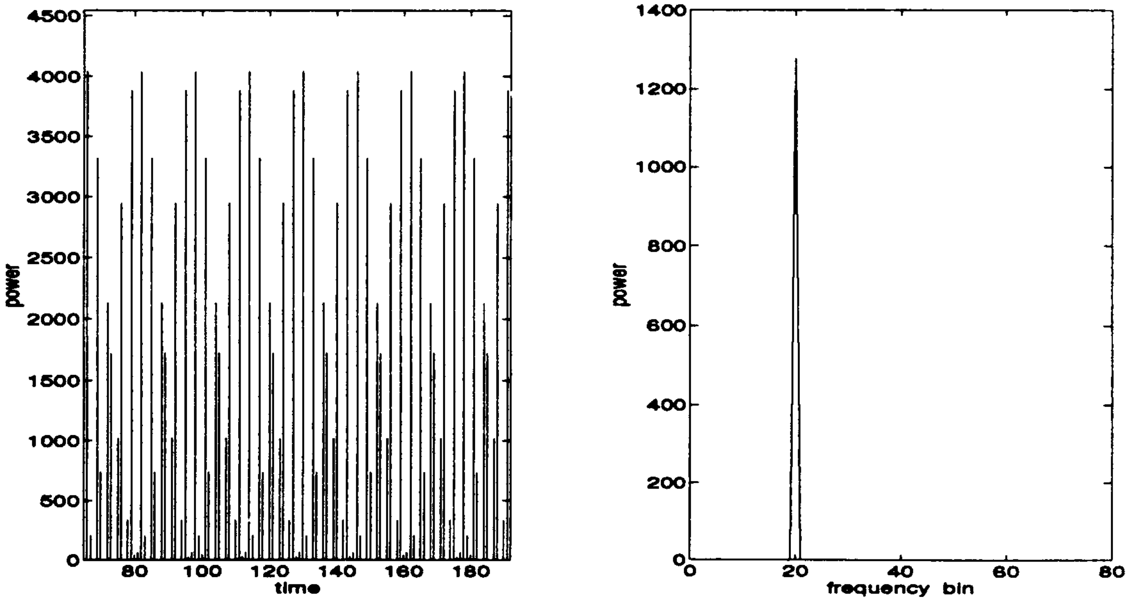
Results of computer simulations using SPECTRO and ONE_HALF validate the theoretical calculations for the periodogram and 1-1/2 D_{ips}, respectively. For the simulations, a 128 point data window is stepped through a 256 point data sequence one point at a time. Both methods apply a rectangular window function to the data. The first and last *winlen*/2 time steps are discarded before computing any averages or plotting any results. The 1-1/2 D_{ips} estimate is squared to allow a fair comparison with the spectrogram. Figure 3.9 shows the mesh plot and corresponding contour plot of the 1-1/2 D_{ips} estimate for the real signal. The sinusoidal signal is centered in frequency bin 20. Bin 20 and the average of each frequency bin are plotted in Figure 3.10(a) and Figure 3.10(b), respectively. The maximum power at any one time step is approximately 4042. This agrees fairly well with the magnitude squared value of (3.11) which is equal to half of the window length squared, or 4096. The intra-ridge modulation exhibited by bin 20 in Figures 3.9 and 3.10(a) results from the cosine cubed modulating term in (3.11). The average power of bin 20 is approximately 1280 as shown in Figure 3.10(b).

Figures 3.11 and 3.12 use the same arrangement of plots to display spectrogram results for the real signal. The maximum power agrees exactly with the theoretical value of 4096 calculated using (3.12). Figure 3.12 shows that the average power of bin 20 is equal to the maximum power. The maximum power of each method is about the same, however, the average power of 1-1/2 D_{ips} is approximately one-third as much as the spectrogram's average power due to the intra-ridge modulation effect.



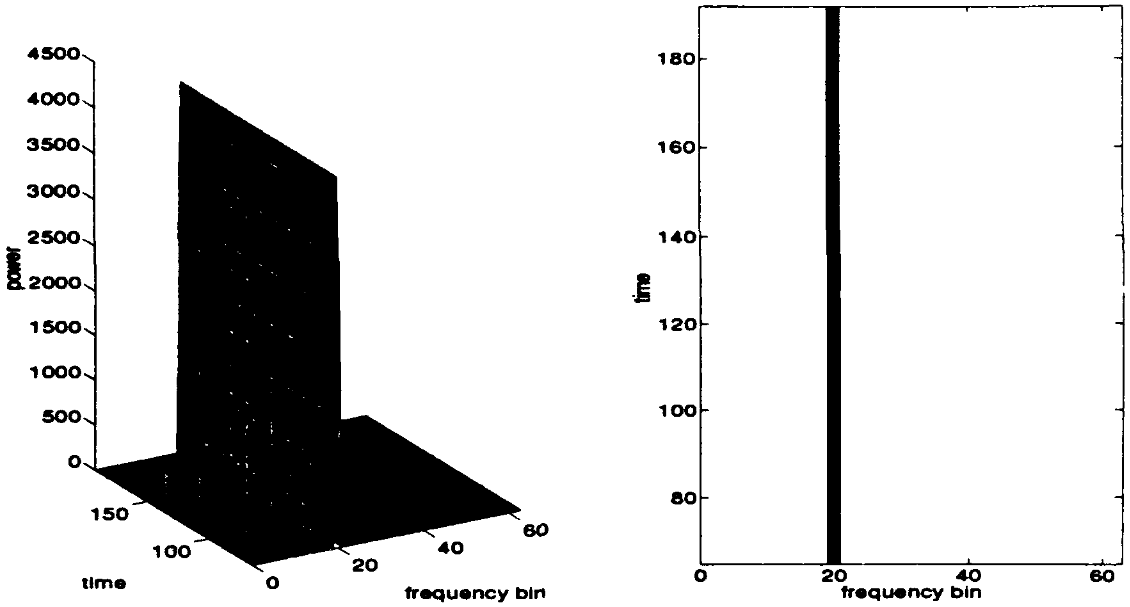
(a) (b)

Figure 3.9: 1-1/2 D_{ips} time-frequency representation of a real sinusoid.



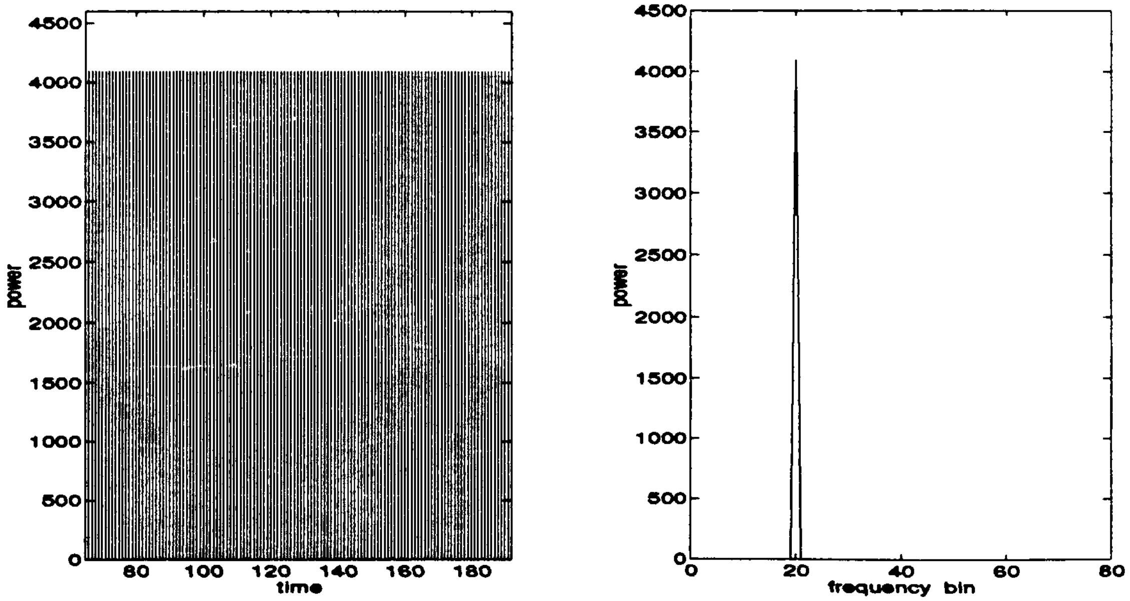
(a) (b)

Figure 3.10: 1-1/2 D_{ips} signal power in bin 20 (real sinusoid).



(a) (b)

Figure 3.11: Spectrogram time-frequency representation of a real sinusoid.



(a) (b)

Figure 3.12: Spectrogram signal power in bin 20 (real sinusoid).

When the same simulations are run with a complex sinusoidal input, ONE_HALF produces a maximum signal power approximately equal to the length of the data window squared which agrees with the squared magnitude of (3.14). As a result, ONE_HALF's maximum power is about equal to the spectrogram's for both real valued and complex valued signals. However, for the complex valued sinusoid the average power in bin 20 of the 1-1/2 D_{ips} estimate is about half that of the spectrogram's rather than a third as it was for the real sinusoid. This is a consequence of the different modulating terms in (3.11) and (3.14). The real signal is modulated by a cosine cubed function which has a magnitude squared average value of about 0.345 for a unit amplitude cosine over one period. In contrast, the complex valued signal is modulated by a cosine function having a magnitude squared average value of about 0.52.

D. SNR PROCESSING GAIN COMPARISONS

The ability of a technique to detect a signal in noise is measured in two ways. The first measure is a processor signal-to-noise ratio (PSNR), or simply the ratio of processor signal power to processor noise power:

$$PSNR = \frac{P_{sig}}{P_n}; \tag{3.16}$$

where P_n is the noise variance. The second approach, denoted PDIF, measures the power of the signal in terms of the number of noise standard deviations above the mean noise power level:

$$PDIF = \frac{P_{sig} - \mu_n}{\sigma_n}; \tag{3.17}$$

where μ_n is the processed noise mean power and σ_n is the standard deviation of the processor output noise power. The ratio of either (3.16) or (3.17) for two processing techniques forms a figure of merit used to compare methods. To compare 1-1/2 D_{ips} and the spectrogram using (3.16) the processing gain ratio is defined as:

$$PG_{SNR} = \frac{PSNR_{ONEHALF}}{PSNR_{SPECTRO}}. \tag{3.18}$$

When (3.17) is the measure of interest, the processing gain ratio is given by:

$$PG_{DIF} = \frac{PDIF_{ONEHALF}}{PDIF_{SPECTRO}}. \tag{3.19}$$

Computer simulations are used to measure the processing gain for various window lengths and input noise variances. The input consists of a fixed amplitude real sinusoid centered in frequency bin 20 and mixed with Gaussian white noise. For this scenario the input signal-to-noise ratio (SNR) is given by:

$$\begin{aligned} SNR &= 10\log\left(\frac{P_{sig}}{P_n}\right) \\ &= 10\log\left(\frac{1}{2\sigma_n^2}\right). \end{aligned} \tag{3.20}$$

Fifteen realizations of each simulation are averaged in order to obtain representative results. As before, the input data sequence is twice the length of the data window, a rectangular window is applied to the data, and the data window is stepped through the input sequence one point at a time. Figure 3.13 shows the resulting plots for a data window length of 128, and an input SNR of -3dB. The fifteen-realization averaged 1-1/2 D_{ips} time-frequency surface is shown in Figure 3.13(a) and its corresponding frequency bin average is displayed in Figure 3.13(b). Figures 3.13(c) and 3.13(d) are the averaged spectrogram time-frequency surface, and its frequency bin averages, respectively. Figures 3.14, 3.15, 3.16, 3.17, and 3.18 display the results for the same window length but for different input SNRs of about -10dB, -15dB, -17dB, -18.5 dB, and -19.5 dB, respectively.

Processing gain comparisons using (3.18) and (3.19) were made of the frequency bin averages as a function of SNR for the signals shown in Figures 3.13 through 3.18. The average value of the signal bin was used for P_{sig} to compute $PSNR$ and $PDIF$. The mean, standard deviation, and variance of the noise were computed using all bins except the one containing the signal. Table 3.3 summarizes the results of these processing gain computations.

The PG_{SNR} performance measure is consistent with results obtained from the noise only simulations and the signal only simulations. As expected from these earlier simulations, PG_{SNR} is greater than one (indicating that 1-1/2 D_{ips} performs better than the spectrogram) when the input variance is smaller than about 1.5. This is agrees with the breakpoint determined from the analysis of Figure 3.8 earlier. Based upon the time-frequency representations however, PG_{DIF} appears to be a more useful indicator of detection performance over a much wider range of input SNRs.

The PG_{DIF} column of Table 3.3 shows that the 1-1/2 D_{ips} frequency bin average signal peak estimate, when measured in terms of noise standard deviations above the mean noise level, is about 1.6 times higher than it is for the spectrogram estimate when the input SNR is about -3dB. As input SNR decreases the relative processing gain also decreases. Both methods are equivalent when input SNR is about -19dB. The spectrogram becomes the better estimate by this standard at SNRs lower than -19dB for this window length.

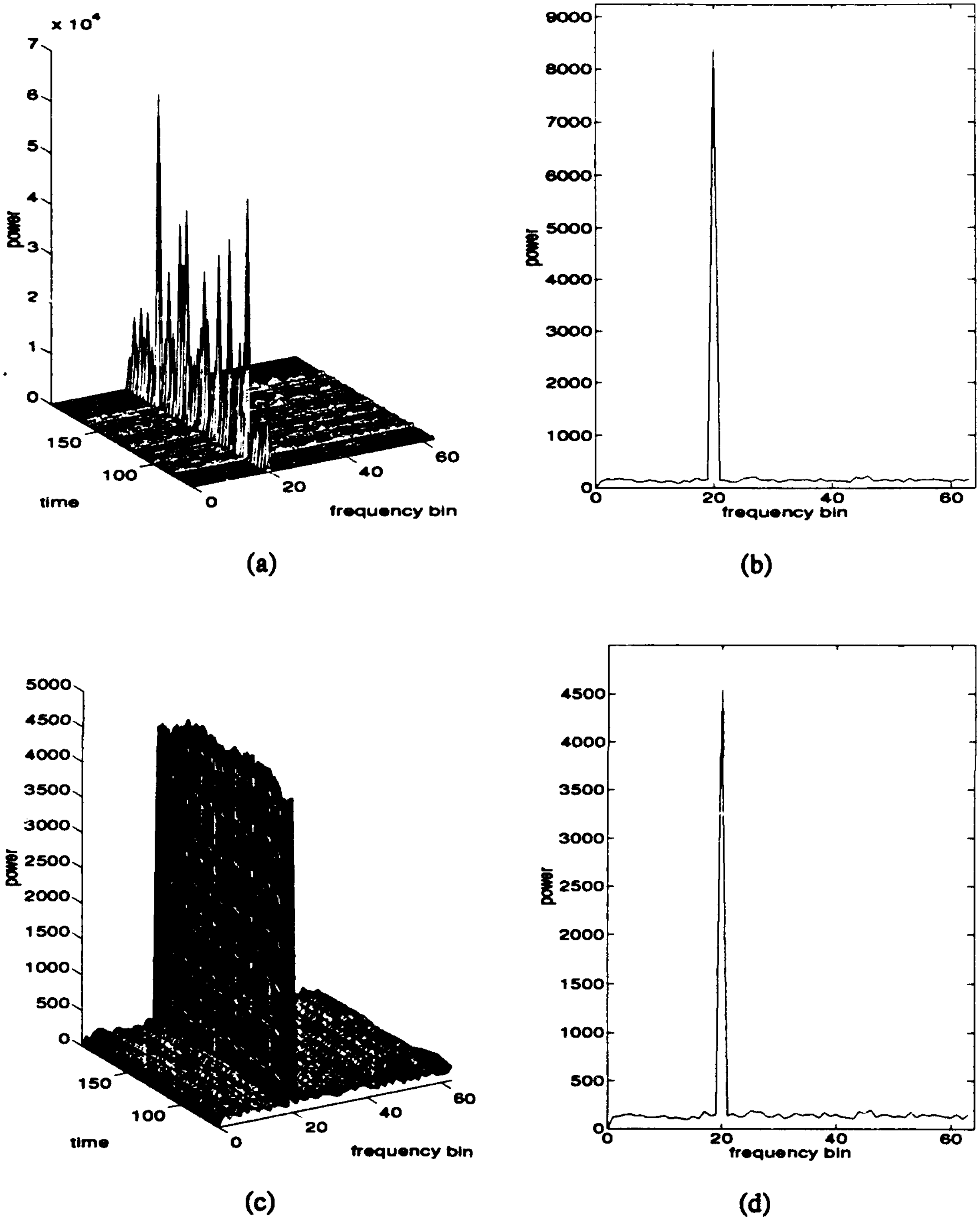
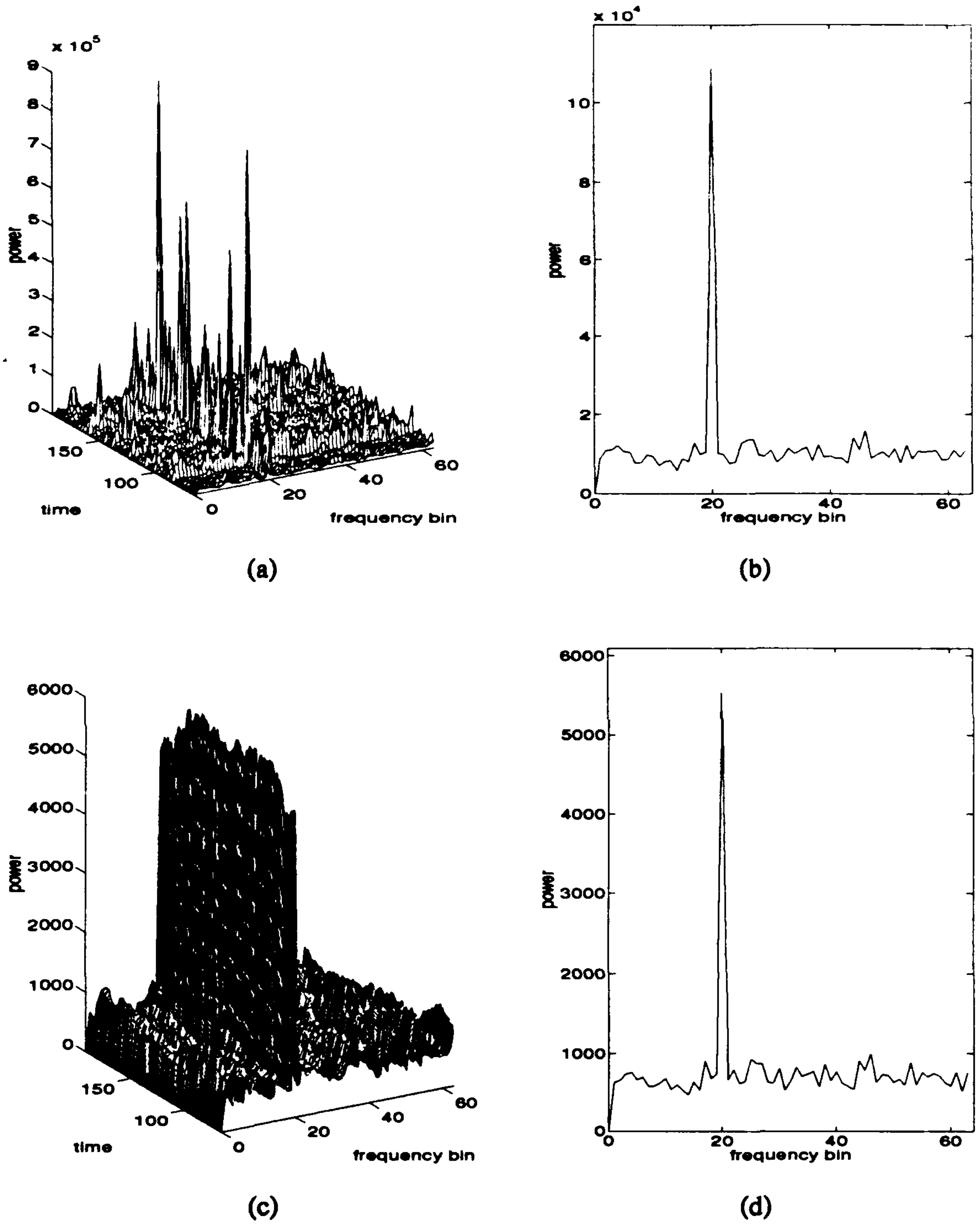


Figure 3.13: Sinusoid in bin 20 with SNR ≅ -3dB; (a) 1-1/2 D_{ips} time-frequency representation, (b) 1-1/2 D_{ips} bin averages, (c) spectrogram time-frequency representation, and (d) spectrogram bin averages.



(a)

(b)

(c)

(d)

Figure 3.14: Sinusoid in bin 20 with SNR ≅ -10dB; (a) 1-1/2 D_{ips} time-frequency representation, (b) 1-1/2 D_{ips} bin averages, (c) spectrogram time-frequency representation, and (d) spectrogram bin averages.

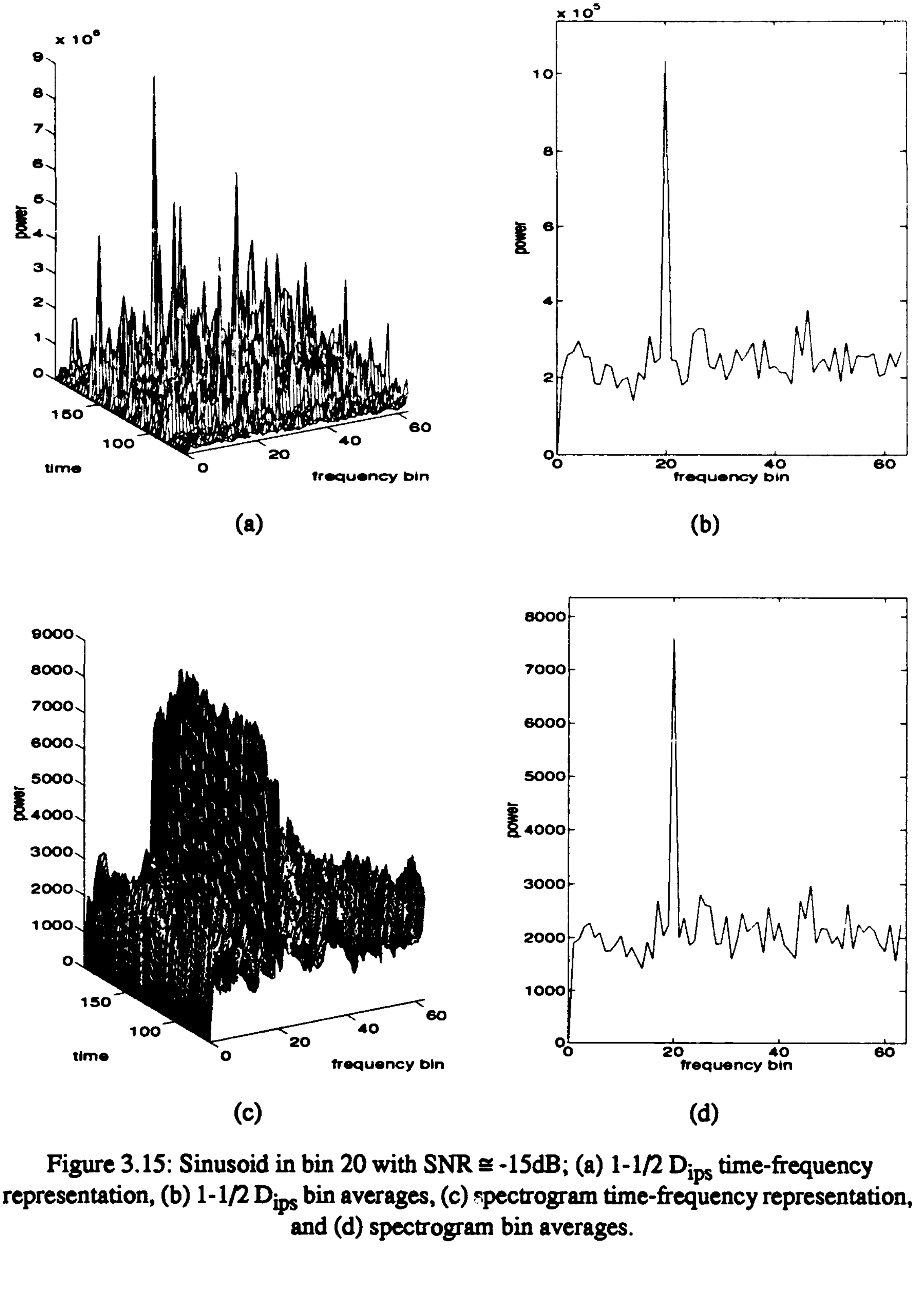


Figure 3.15: Sinusoid in bin 20 with SNR ≅ -15dB; (a) 1-1/2 D_{ips} time-frequency representation, (b) 1-1/2 D_{ips} bin averages, (c) spectrogram time-frequency representation, and (d) spectrogram bin averages.

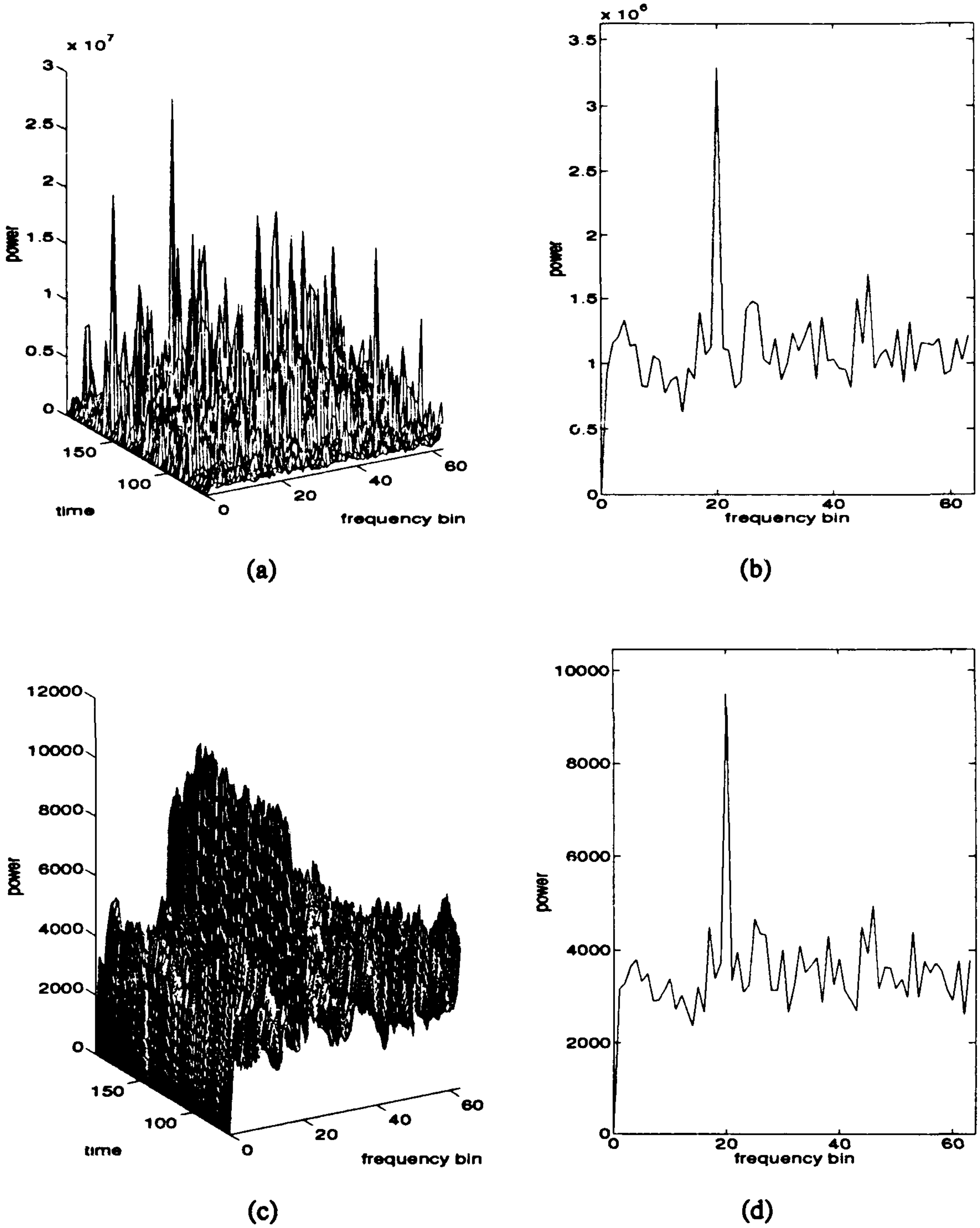
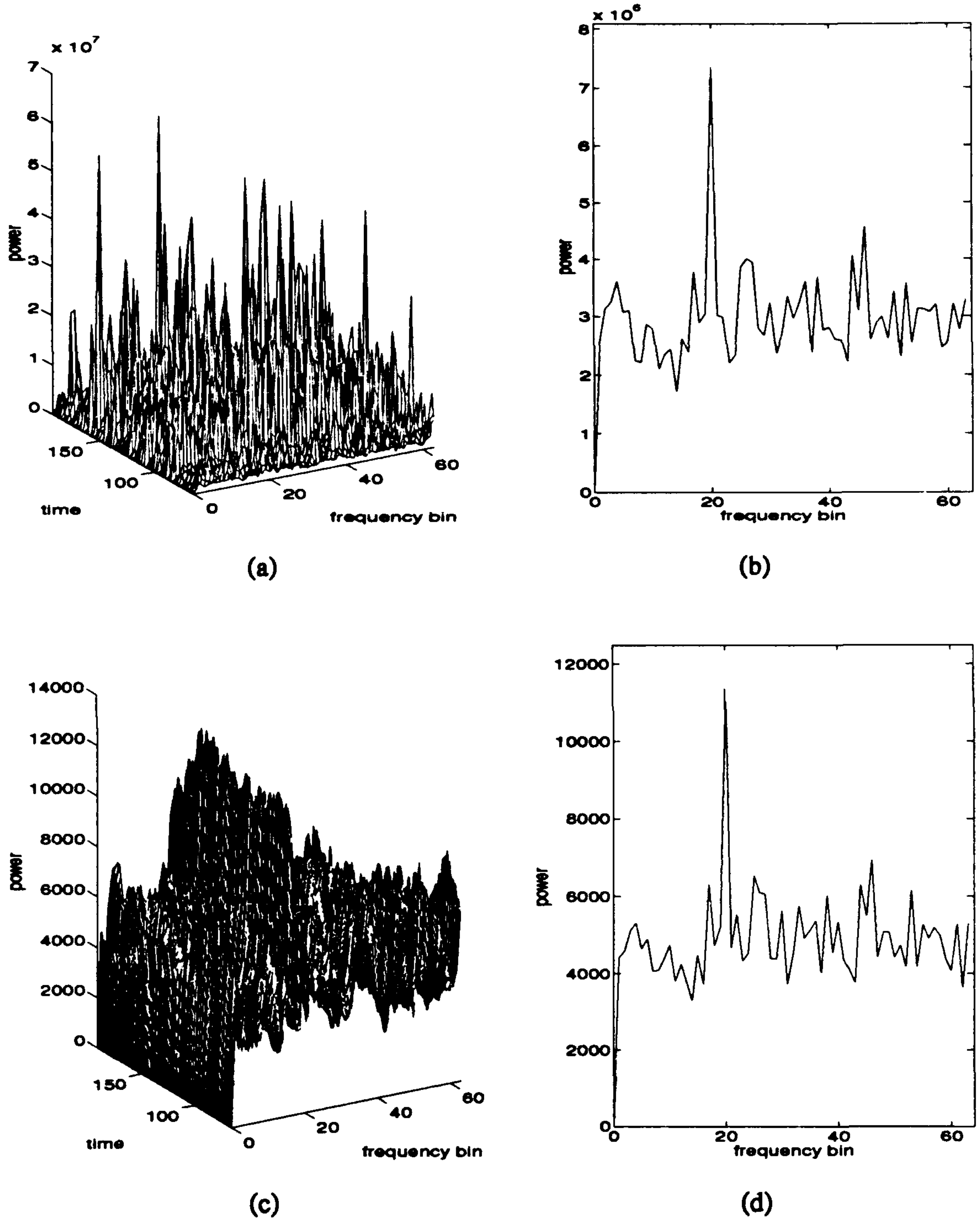


Figure 3.16: Sinusoid in bin 20 with SNR ≅ -17dB; (a) 1-1/2 D_{ips} time-frequency representation, (b) 1-1/2 D_{ips} bin averages, (c) spectrogram time-frequency representation, and (d) spectrogram bin averages.



(a)

(b)

(c)

(d)

Figure 3.17: Sinusoid in bin 20 with SNR ≅ -18.5dB; (a) 1-1/2 D_{ips} time-frequency representation, (b) 1-1/2 D_{ips} bin averages, (c) spectrogram time-frequency representation, and (d) spectrogram bin averages.

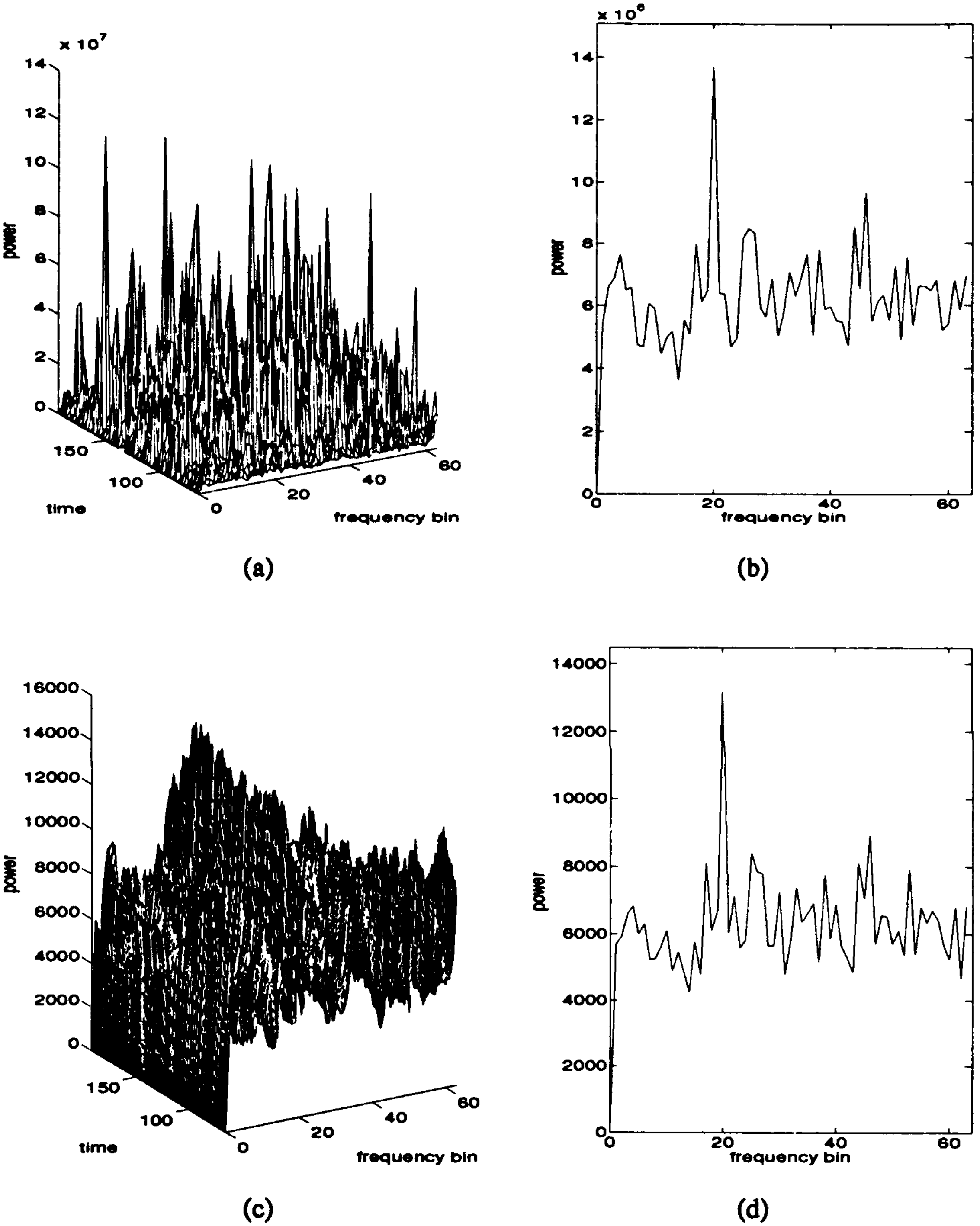


Figure 3.18: Sinusoid in bin 20 with SNR ≅ -19.5dB; (a) 1-1/2 D_{ips} time-frequency representation, (b) 1-1/2 D_{ips} bin averages, (c) spectrogram time-frequency representation, and (d) spectrogram bin averages.

TABLE 3.3: RELATIVE PROCESSING GAINS FOR VARYING INPUT SNR AND A FIXED DATA WINDOW LENGTH OF 128

input SNR (-dB)	PG_{SNR}	$PDIF_{ONEHALF}$	$PDIF_{SPECTRO}$	PG_{DIF}	corresponding figure no.
3.01	1.4270	217.8826	132.4518	1.6450	3.12
10.00	< 1	37.9275	29.3067	1.2942	3.13
14.77	< 1	12.8635	11.2700	1.1414	3.14
16.99	< 1	8.0222	7.4846	1.0718	3.15
18.45	< 1	5.9961	5.8097	1.0269	3.16
19.54	< 1	4.8344	4.8552	0.9957	3.17

Results of signal-in-noise simulations run for a smaller 64 point data window and a larger 256 point data window are consistent with those obtained for the 128 point window. However, the range of input SNRs over which the 1-1/2 D_{ips} method yields a larger processing gain than does the spectrogram depends upon window length. For the 64 point window, 1-1/2 D_{ips} yields a larger processing gain than the spectrog ım only for input SNRs greater than about -12.5dB. As already demonstrated, for a 128 point window 1-1/2 D_{ips} has a larger processing gain than the spectrogram for input SNRs greater than about -19dB. Simulations using a 256 point window showed that 1-1/2 D_{ips} performs better than the spectrogram in terms of processing gain even when the input SNR is less than -21dB and the signal cannot be readily discerned from the noise in the averaged frequency bin plots. In effect, for this type of signal and processing scheme using a 256 point data window, 1-1/2 D_{ips}'s processing gain is between 1.15 and 1.64 times larger than the spectrogram's for input SNRs between -21dB and -3dB. The simulations also indicate that the processing gain performance of 1-1/2 D_{ips} improves relative to the spectrogram with increasing window length.

IV. SIMULATION RESULTS AND ANALYSIS

Simulation results are presented to show how 1-1/2 D_{ips}, the bispectrum methods, IHOMS, and the spectrogram deal with three types of signals. The spectrogram, bispectrum, and 1-1/2 D_{ips} methods ar: first compared using a signal consisting of multiple unrelated stationary sinusoids that are mixed with white Gaussian noise. Next, the bispectral representation of noise free signals containing harmonically related stationary components is studied. Lastly, a received signal containing multiple linear FM components mixed with white Gaussian noise is processed using IHOMS, 1-1/2 D_{ips}, and the spectrogram.

A. STATIONARY SINUSOIDS

The spectrogram is computed using the Matlab function SPECTRO, and the 1-1/2 D_{ips} method is implemented via ONE_HALF. Their time-frequency representations for a single sinusoidal signal are discussed in Chapter III.

The bispectrum methods produce a frequenc; frequency representation instead of a time-frequency representation. The direct method for computing the bispectrum is realized through the Hi-Spec Matlab function BISPEC_D of [Ref. 14]. This function returns the complex bispectrum matrix and a frequency axis labeling vector. The user specifies an input data sequence, the desired Fast Fourier Transform (FFT) size, the number of samples per segment, the amount of overlap between data segments, and the desired form of the frequency smoothing window. The extrinsic Matlab function INDBIS contained in Appendix C implements the indirect method of calculating the bispectrum. It returns the complex bispectrum of the signal supplied by the user. The FFT size, the number of data samples per segment, and the desired tri-correlation window function are specified by the user. The three window options available are the unit hexagonal window, the Parzen window, and the optimum window (also known as the minimum bias supremum window).

Both bispectrum methods are used to generate bispectral representations of a noise free sinusoid located at frequency bin 20. The input sequence consists of 256 samples. Both methods apply a 128 puint FFT to each of four non-overlapping data segments containing 64 data points each. The indirect method representations are formed from tri-correlation functions based upon 31 lags. Figure 4.1 shows the full bispectrum plane representation of a real sinusoid generated by the direct method without a window The sinusoid is indicated by the peak in the first quadrant located at $(k_1, k_2) = (20, 20)$, and associated symmetry peaks like the one in the third quadrant at $(k_1, k_2) = (-20, -20)$.

The representation of a real sinusoid at bin 20, formed by the indirect method using a unit hexagonal window is displayed in Figure 4.2. The rectangular window used by both ONE_HALF and SPECTRO, the unit hexagonal window used by INDBIS, and the unwindowed BISPEC_D are essentially equivalent window functions so that representations formed using them are considered unwindowed. Two closely spaced peaks are discernible at the signal locations in the indirect method's representation. This is a characteristic of this method which depends upon the number of lags used to compute the tri-correlation function. The two peaks at each location become less distinct as the number of lags used to compute the tri-correlation sequence is increased. Another difference between Figure 4.1 and 4.2 is that the indirect representation exhibits peaks at $(k_1, k_2) = (20, -20)$ and $(k_1, k_2) = (-20, 20)$ which are not noticeable in the direct method's representation. It turns out that both of the methods are sensitive to the phase of a real signal, and that the methods produce slightly different representations for the same phase. The signal used to create the bispectrum in Figure 4.1 has a different phase than the signal which the bispectrum in Figure 4.2 is based upon. Both phase values are specially chosen to chosen to minimize any peaks on the $k_1 = 0$ and $k_2 = 0$ axes. In the case of the direct method's representation, this also minimized the peaks at $(k_1, k_2) = (20, -20)$ and $(k_1, k_2) = (-20, 20)$. Bispectral peaks located on either the k_1 or the k_2 axis in subsequent plots are referred to as zero axis peaks.

The unwindowed full bispectrum representations for a complex sinusoid obtained by the direct method and indirect method are shown in Figure 4.3 and 4.4, respectively. Like the real signal representation, the complex signal produces a peak at $(k_1, k_2) = (20, 20)$. The complex signal's bispectrum does not exhibit as many symmetry peaks as does the real signal's representation because the bispectrum of a complex signal is symmetric about only one symmetry line and in only one quadrant. Which line, and which quadrant are determined by the conjugation scheme used to compute the bispectrum [Ref. 9]. For the equivalent schemes used by INDBIS and BISPEC_D which are described in Chapter II, symmetry exists about the $k_1 = k_2$ line in the first quadrant. The peaks at $(k_1, k_2) = (0, 20)$ and $(k_1, k_2) = (20, 0)$ in both representations are zero axis peaks. Unlike real signal representations, zero axis peaks in the bispectrum of a complex signal have a consistent amplitude regardless of signal phase.

The first bispectral quadrant is essentially the same for either a real or a complex sinusoid, and is sufficient for measuring a sinusoidal signal's frequency. The only additional information not contained in the first quadrant is whether or not the signal is complex or real. The remaining bispectrum figures in this thesis will display just the first quadrant.

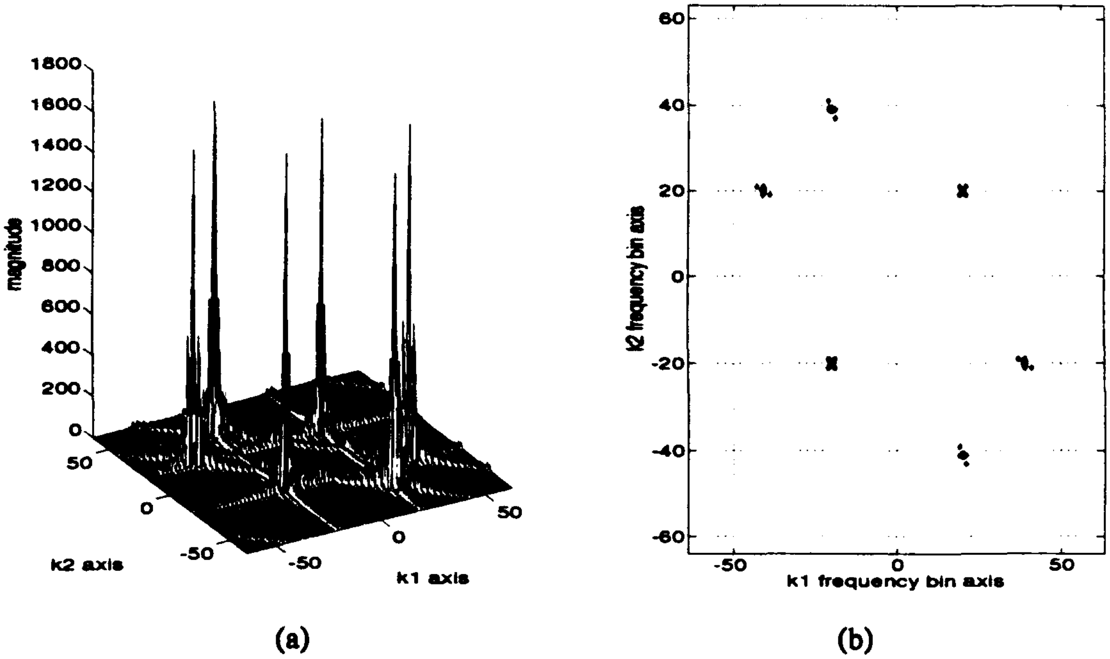


Figure 4.1: Full bispectrum of a real sinusoid at bin 20 (unwindowed direct method).

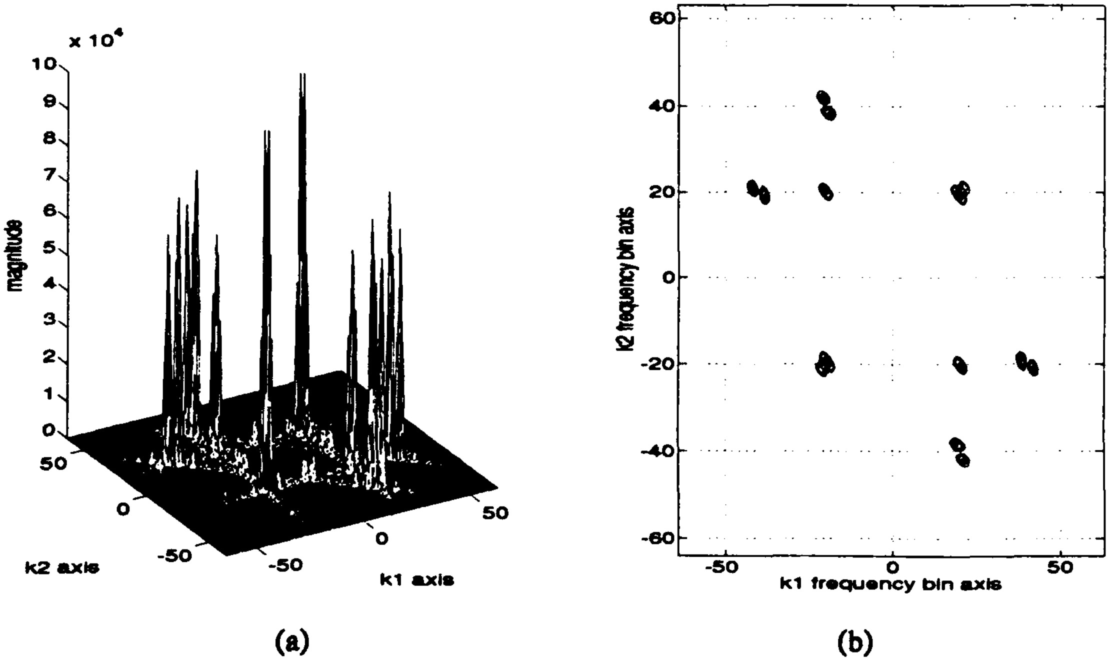
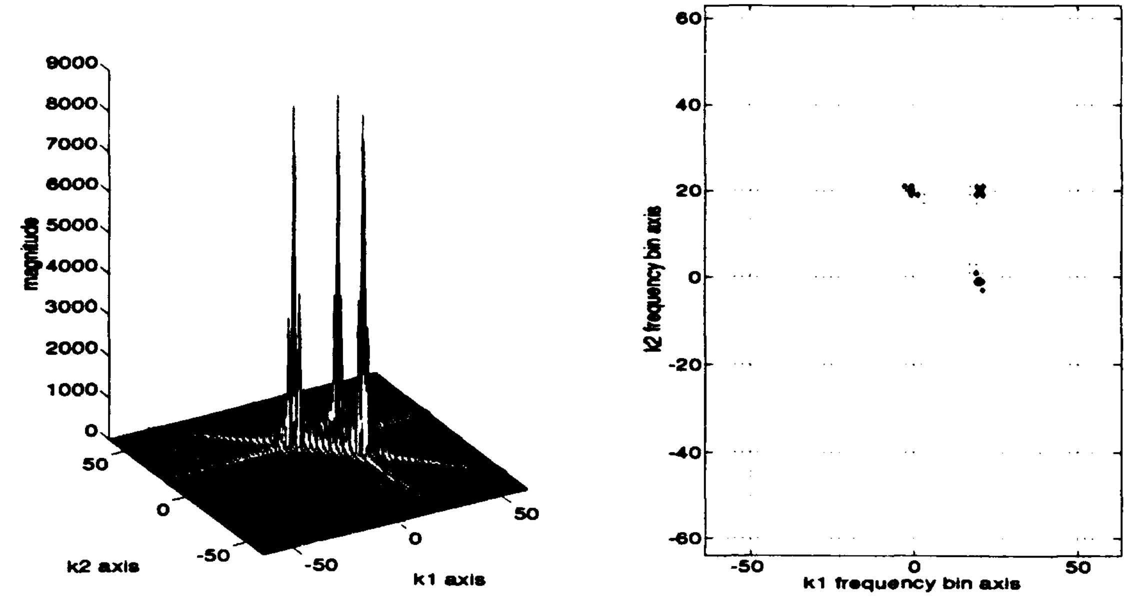
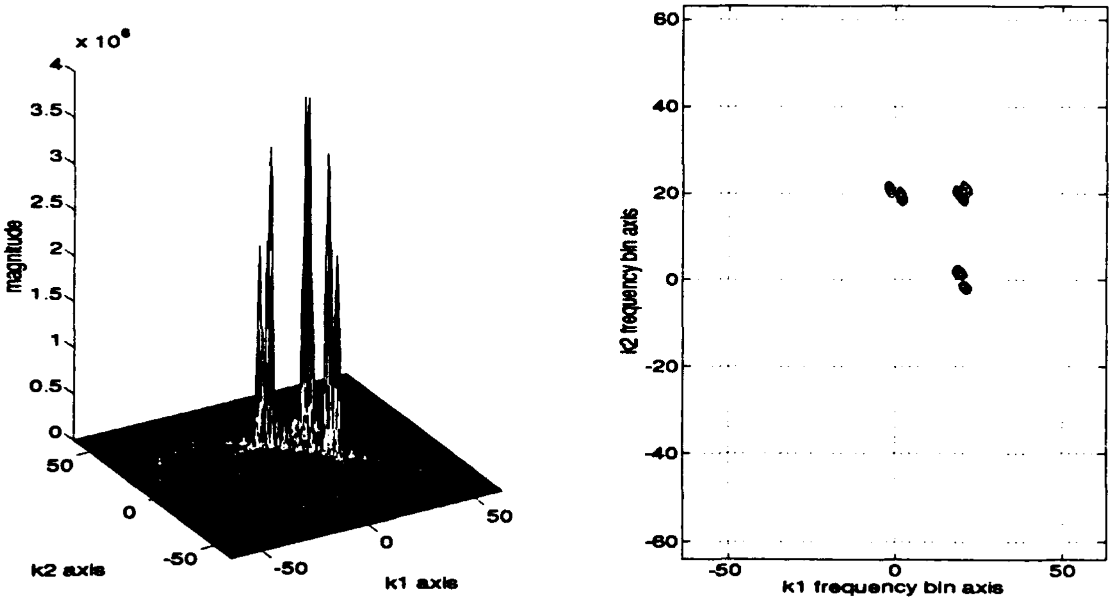


Figure 4.2: Full bispectrum of a real sinusoid at bin 20 (unwindowed indirect method).



(a) (b)

Figure 4.3: Full bispectrum of a complex sinusoid at bin 20 (unwindowed direct method).

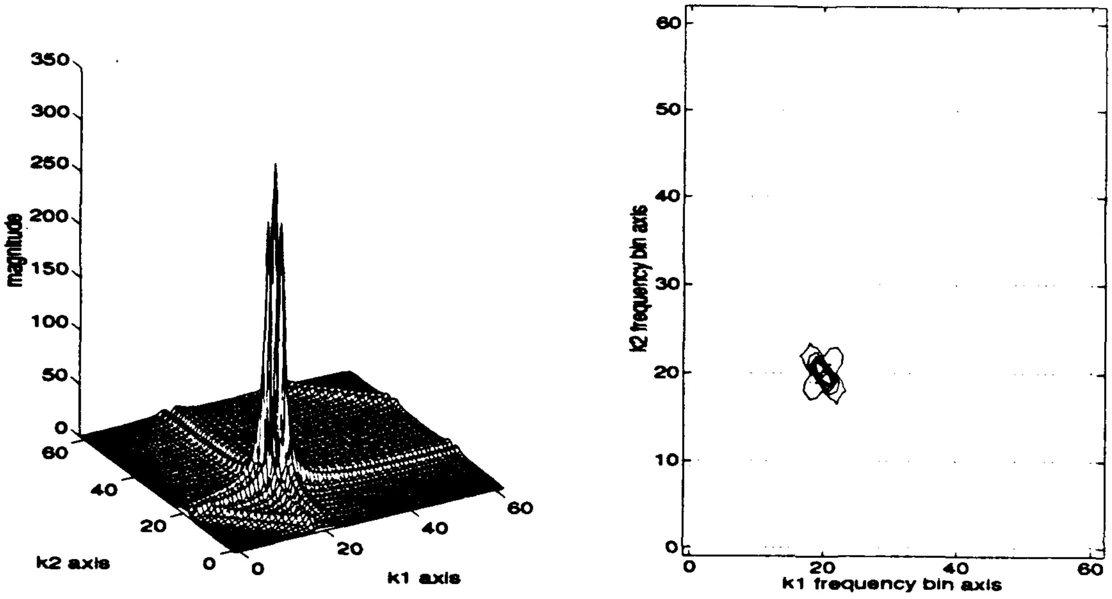


(a) (b)

Figure 4.4: Full bispectrum of a complex sinusoid at bin 20 (unwindowed indirect method).

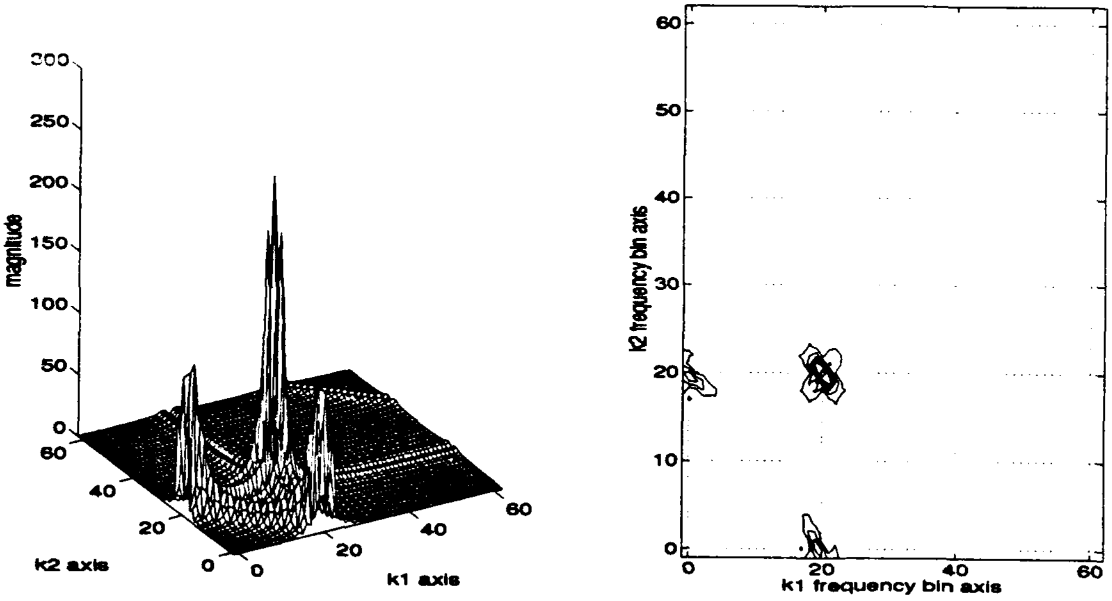
The next four examples of single real valued sinusoid bispectrums are intended to show what windowing can do for each method. At the same time, the effects of signal phase on the bispectral representations are noted. Figure 4.5 shows the bispectrum as computed by the direct method and smoothed by a Rao-Gabr window over 5^2 adjacent frequency points. The signal is a real sinusoid located at bin 20 with a phase angle of 0.484 radians. The same processing scheme yields the representation shown in Figure 4.6 if the signal phase is changed to 4.3151 radians. The indirect method, employing a Parzen window on a real signal with a phase equal to 0.484 radians produces the representation shown in Figure 4.7. When the signal with a phase of 4.3151 radians is processed by the indirect method using an optimum window (the optimum window and Parzen window are not significantly different for this type of signal), it produces the bispectrum shown in Figure 4.8. Windowing helped to make the two distinct peaks at $(k_1, k_2) = (20, 20)$ appear as one in the indirect representation. The zero axis peaks of the indirect representation incorrectly appear to be offset from the zero axes because the first quadrant plots do not show the adjacent peaks in the second and fourth bispectral quadrants. The signal frequency is more accurately and more clearly depicted by the direct method's peak locations. In addition, the direct method is faster than the indirect method as implemented in this study. However, if the signal is known to be either complex or real, the indirect method can be implemented in such a way as to take advantage of the bispectrum symmetry in order to reduce computational costs. This is done in [Ref. 14] to implement the indirect method for real signals in Hi-Spec's Matlab function BISPEC_I.

The bispectrum, spectrogram, and 1-1/2 D_{ips} methods are now used to process a signal consisting of three complex valued sinusoids in additive white Gaussian noise. The three frequencies of the signal are at bins 10.65, 28.05, and 54.7. Noise variance is fixed while signal amplitude is varied to achieve different input SNRs. The phase angle of each signal



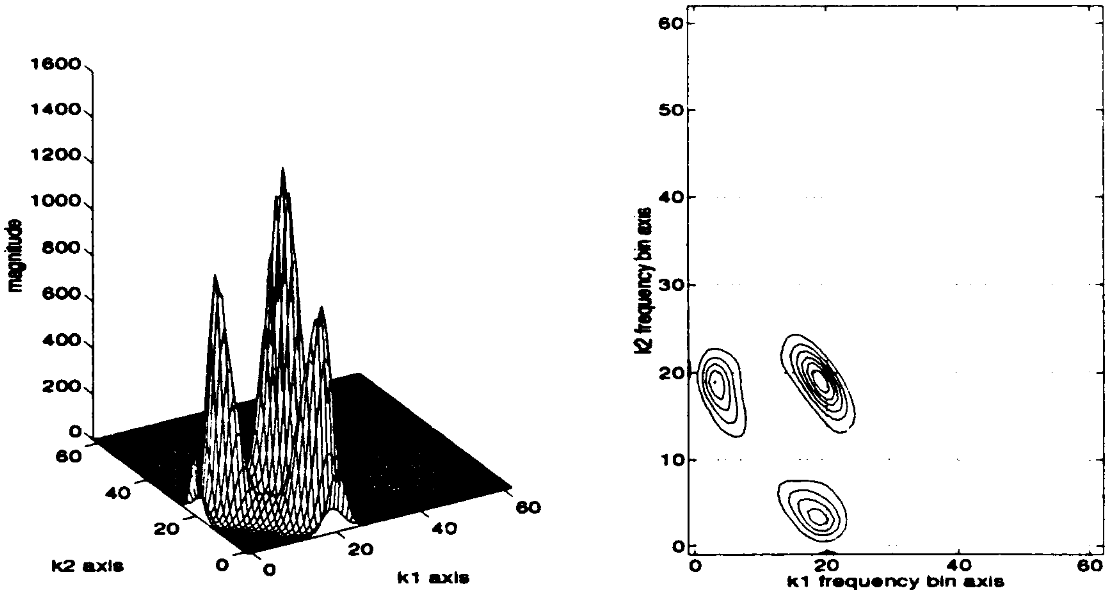
(a) (b)

Figure 4.5: Rao-Gabr smoothed direct bispectrum of real sinusoid (0.484 radians).



(a) (b)

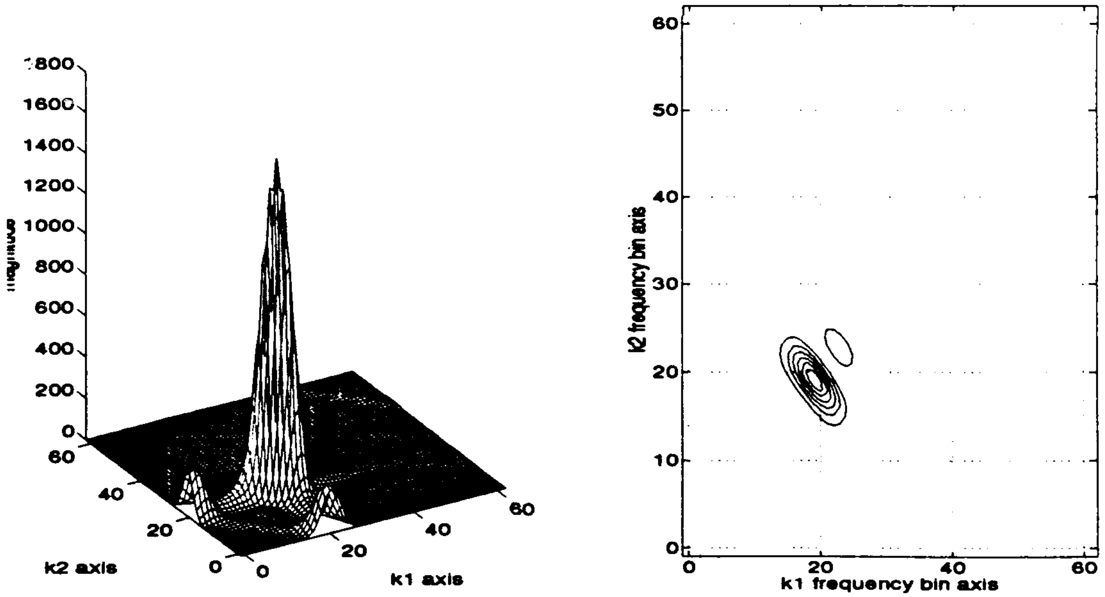
Figure 4.6: Rao-Gabr smoothed direct bispectrum of real sinusoid (4.3151 radians).



(a)

(b)

Figure 4.7: Parzen windowed indirect bispectrum of real sinusoid (0.484 radians).



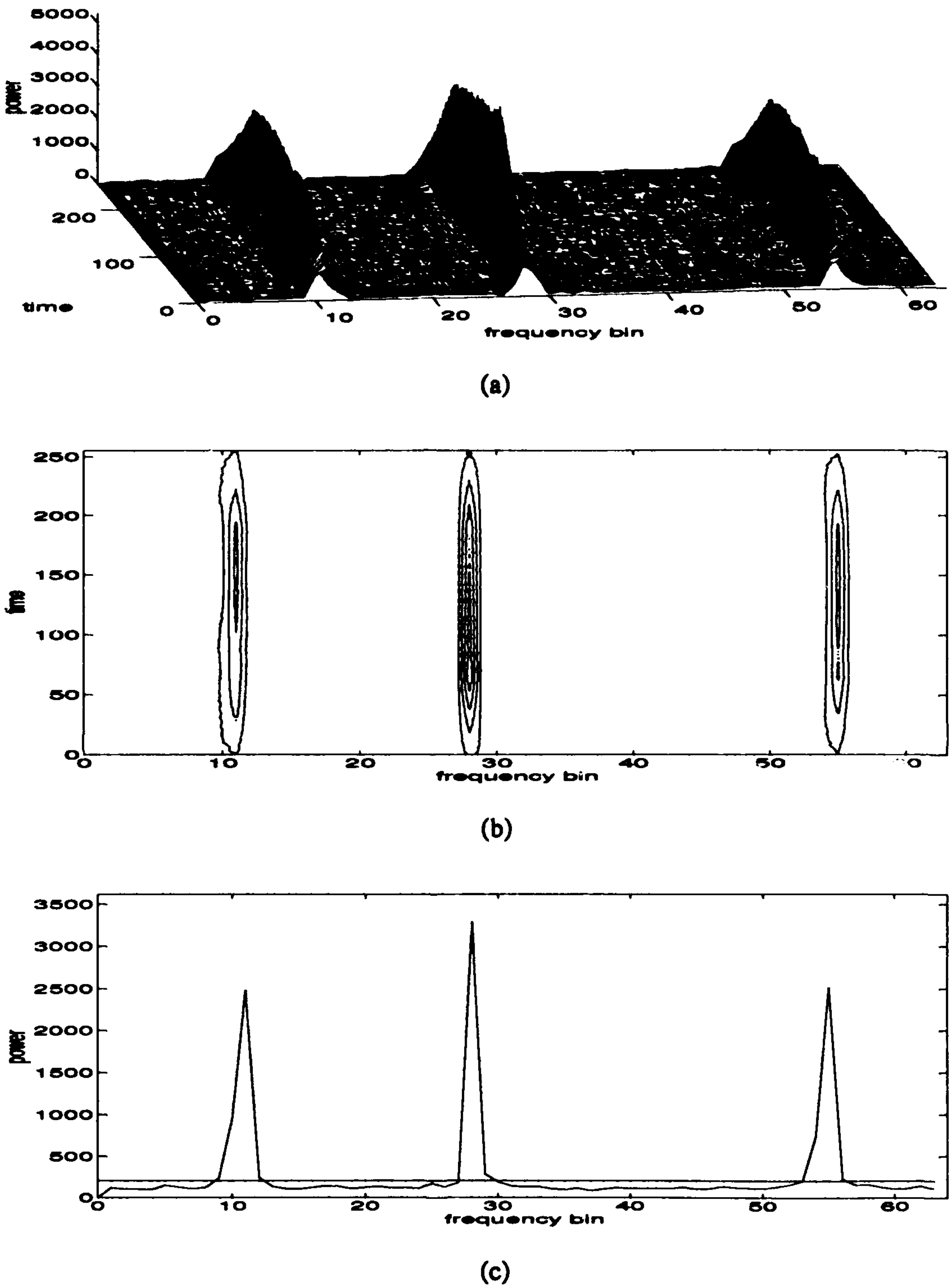
(a)

(b)

Figure 4.8: Optimum windowed indirect bispectrum of real sinusoid (4.3151 radians).

component is distributed uniformly on the interval $(0, 2\pi)$. The data sequence has 256 samples. The 1-1/2 D_{ips} spectrum is computed by ONE_HALF, the spectrogram is generated by SPECTRO, and the bispectrum is calculated by the direct method using BISPEC_D. A step increment of one is employed for both SPECTRO and ONE_HALF, while a 50% overlap is specified for BISPEC_D. To allow for a fair comparison, all three methods are unwindowed. Fifteen representations for each method are computed using different noise realizations. These surfaces are then averaged to obtain either a representative time-frequency surface, or a representative frequency-frequency surface, as appropriate. The averaged surface, along with its frequency bin average is displayed for various input SNRs and window length. The bispectral frequency bin average is the average power in each k_1 frequency bin. A horizontal line on the frequency bin plot is placed three noise standard deviations above the noise mean to provide a reference point. The noise statistics are computed after removing outliers from the frequency bin average values. Figure 4.9 shows the spectrogram representation for a 128 point window length, and an input SNR of -6dB. The 1-1/2 D_{ips} spectrum, and the bispectrum for the same input SNR and window length are displayed in Figures 4.10 and 4.11, respectively. The signal components are clearly detected by all three methods but the symmetry of the bispectrum causes extra peaks that extend above the three standard deviation reference line in Figure 4.11(c).

Figures 4.12, 4.13, and 4.14 show the spectrogram, the 1-1/2 D_{ips} spectrum and the bispectrum, respectively, for an input SNR of -18dB and a window length of 128. At this lower SNR it is difficult to distinguish the signal components in the methods' mesh plots, Figure 4.12(a), 4.13(a), and 4.14(a). However, the sinusoids are discernible in the contour plots and bin average plots of all three methods without any spurious peaks exceeding the three standard deviation reference line. The slight processing gain advantage of the

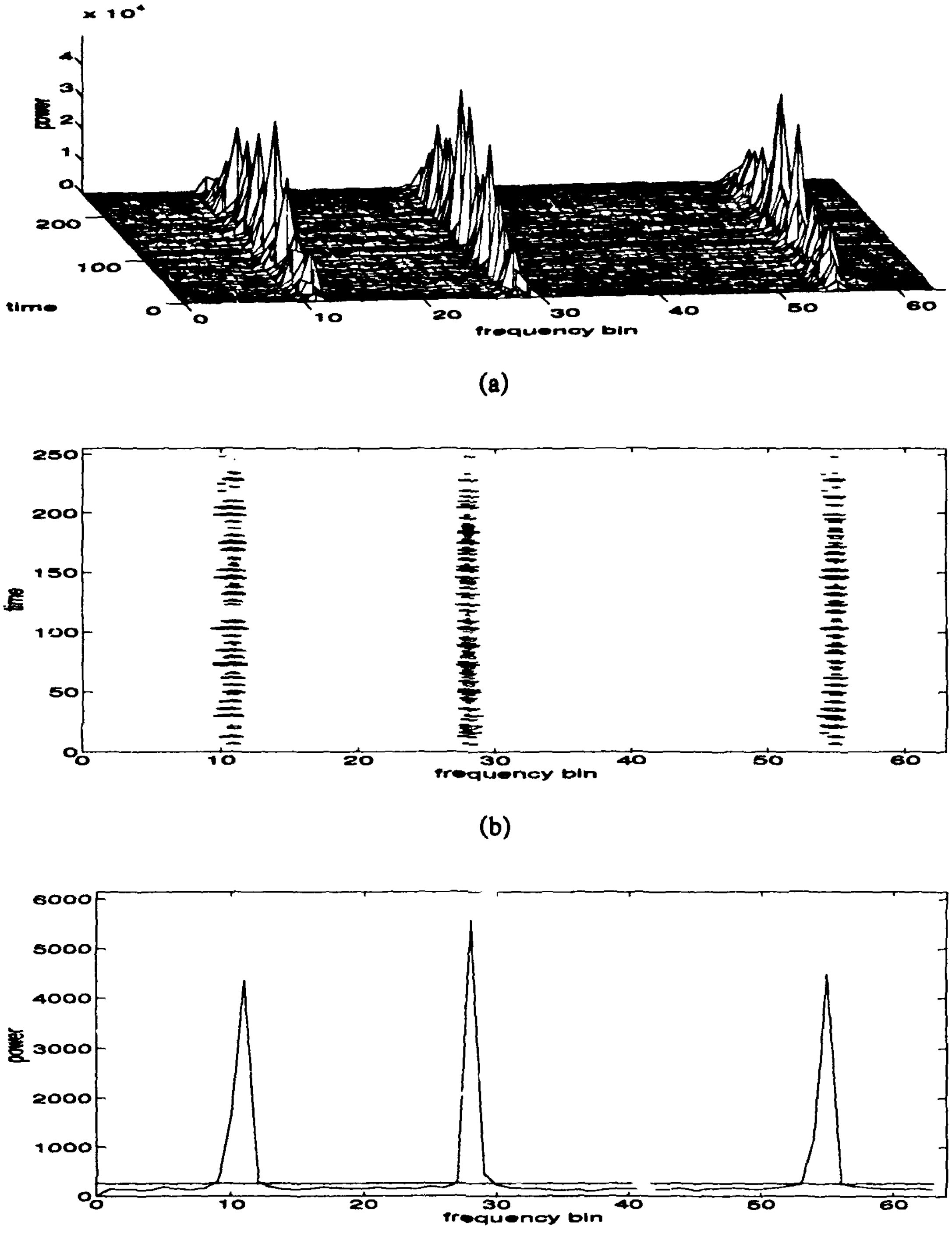


(a)

(b)

(c)

Figure 4.9: Spectrogram of three complex sinusoids at an input SNR of -6dB. The representation is formed using a 128 point data window.

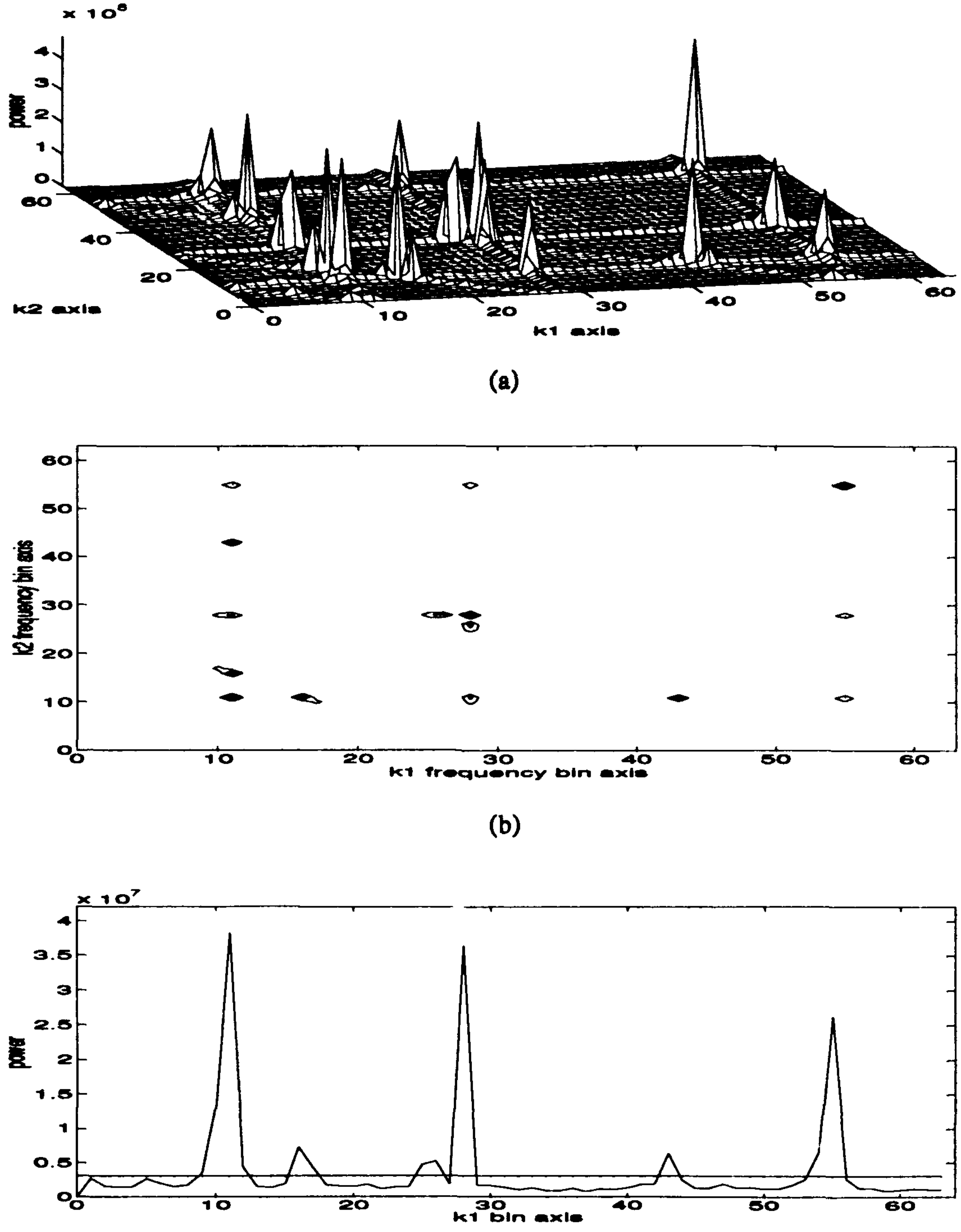


(a)

(b)

(c)_

Figure 4.10: 1-1/2 D_{ips} spectrum of three complex sinusoids at an input SNR of -6dB. The representation is formed using a 128 point data window.



(a)

(b)

(c)

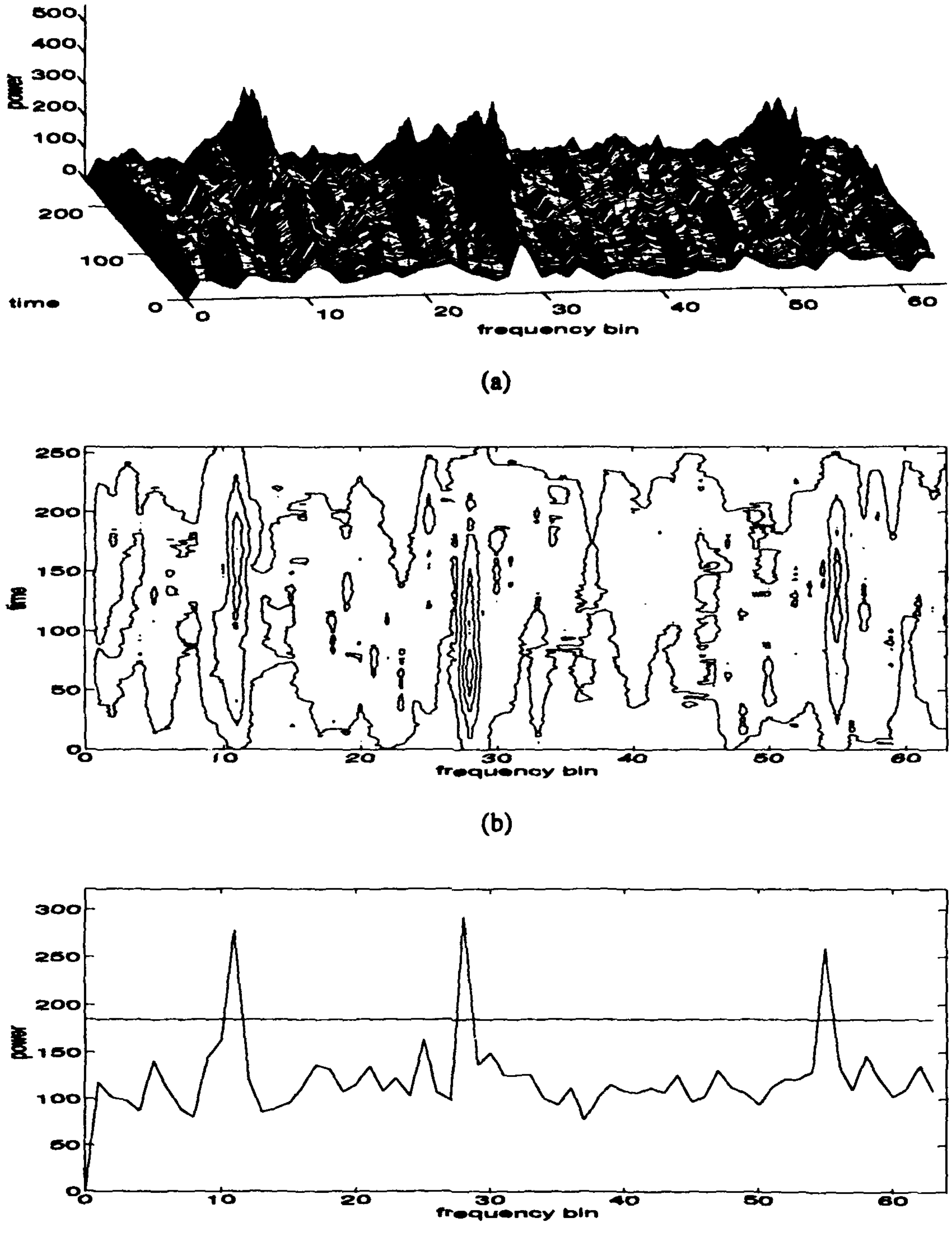
Figure 4.11: Bispectrum of three complex sinusoids at an input SNR of -6dB. The representation is formed using a 128 point data window.

1-1/2 D_{ips} method over the spectrogram at this SNR and window length is barely noticeable in Figures 4.12(c) and 4.13(c). But, the signal peaks and the larger noise peaks in the spectrogram plot are closer to the three standard deviation reference line than they are in the 1-1/2 D_{ips} plot. Relative to the spectrogram, the bispectrum signal peaks have the same height above the reference line but the noise peaks are closer to the line.

Figure 4.15 displays just the frequency bin plots of the three methods for an input SNR of -21dB. The spectrogram has a slight advantage over 1-1/2 D_{ips} at this SNR in terms of the proximity of signal peaks and noise peaks to the three standard deviation line. This is consistent with Chapter III simulation results which indicate that the relative processing gain favors the spectrogram below -19.5dB for this window length. The bispectrum has relatively weaker signal peaks and larger noise peaks than either of the other two methods.

Figure 4.16 shows the frequency bin average plots for the same SNR of -21dB but a larger data window of 256 points. The 1-1/2 D_{ips} spectrum, as expected for this larger window length, appears to perform slightly better than the spectrogram. However, one noise peak at bin 60 in the 1-1/2 D_{ips} plot just reaches the three standard deviation line in Figure 4.16(b). The bispectrum has strong signal peaks and relatively low noise peaks except for the one at about bin 118 that exceeds the reference level.

Figure 4.17 displays the frequency bin plots for an SNR of -26dB and a window length of 256. For all three methods, only the first signal component exceeds the reference level. The 1-1/2 D_{ips} representation's signal peak has a slightly larger relative height above the reference line than either of the other methods. No one method distinguishes itself with respect to suppressing the noise peaks in this case.

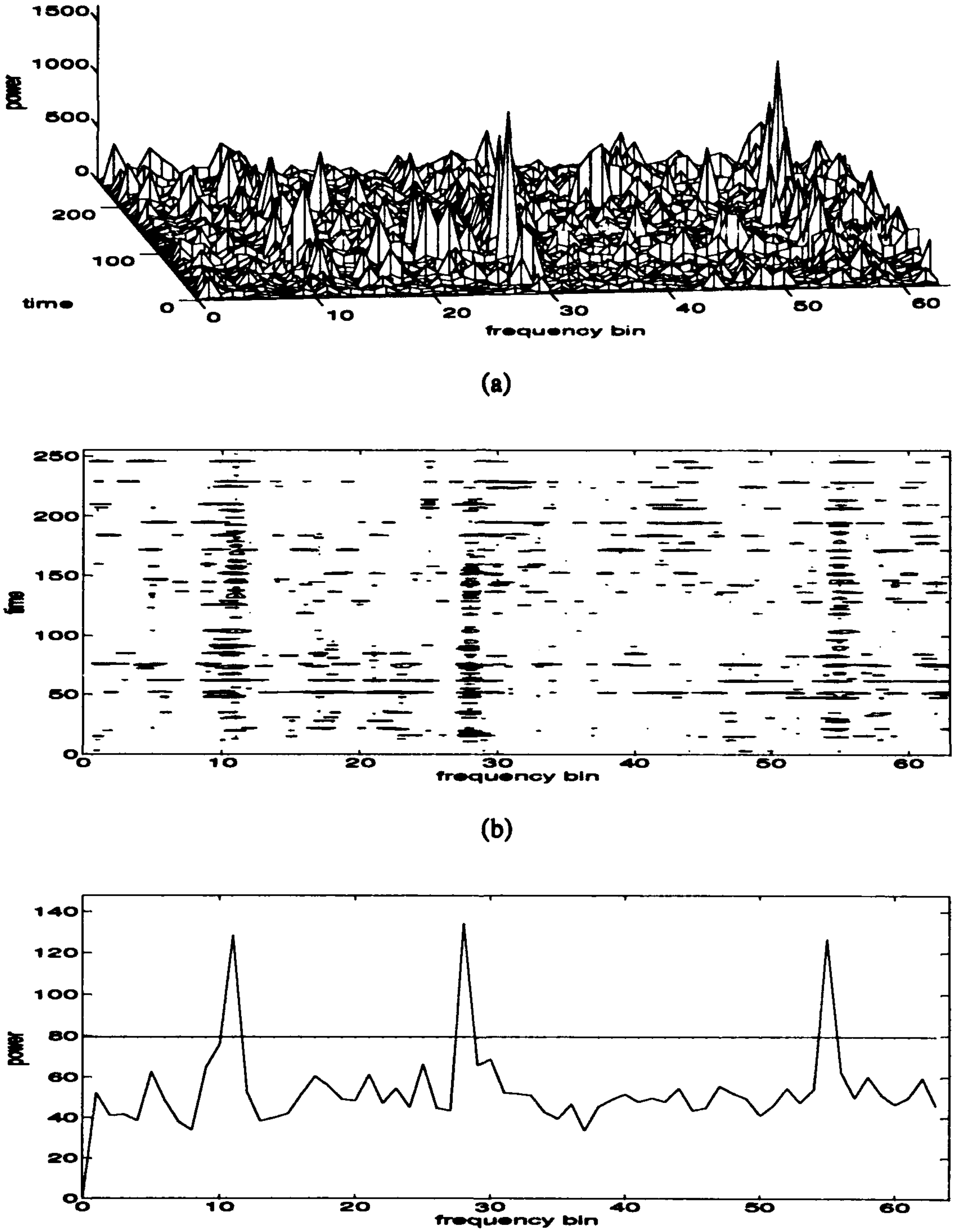


(a)

(b)

(c)

Figure 4.12: Spectrogram of three complex sinusoids at an input SNR of -18dB. The representation is formed using a 128 point data window.

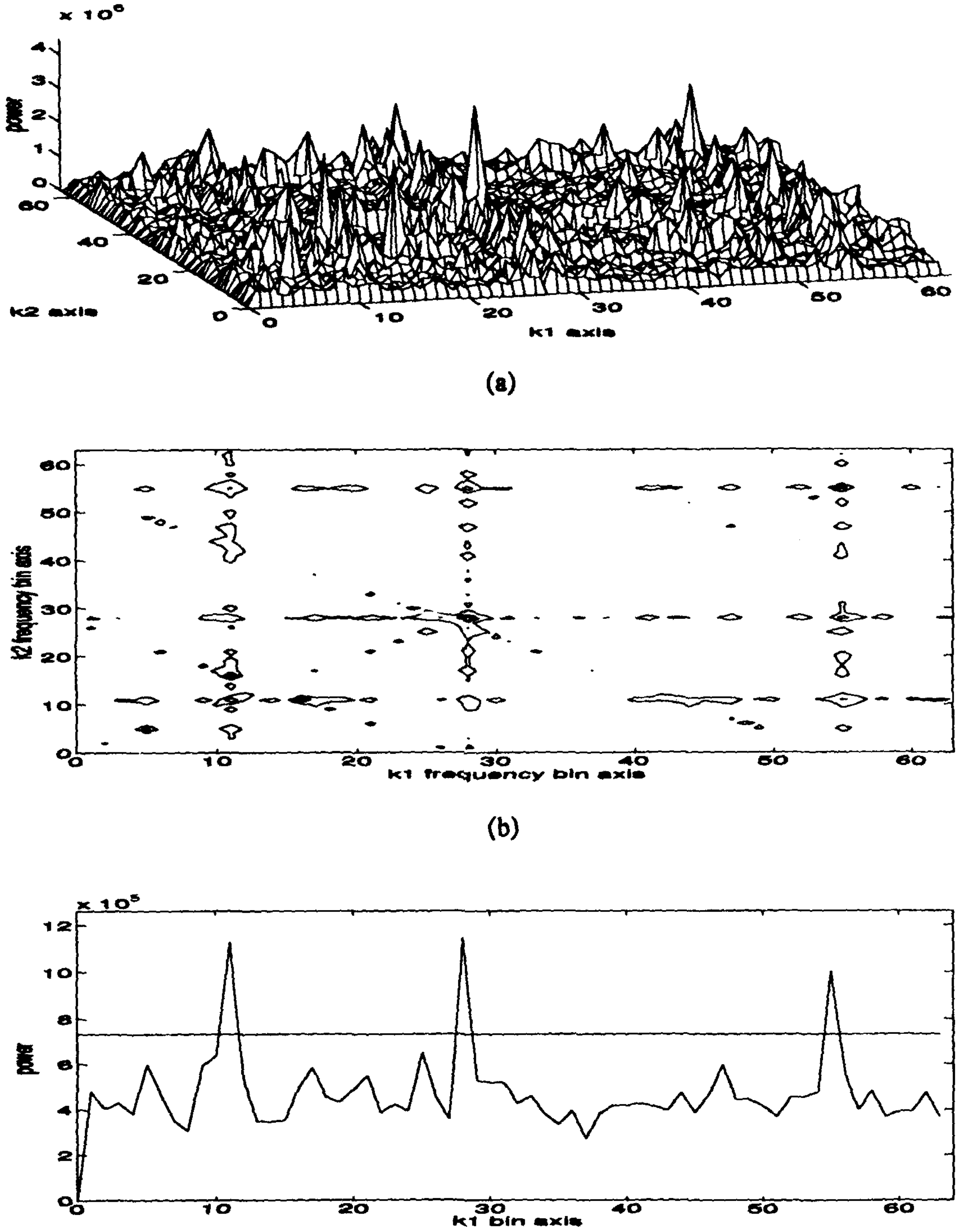


(a)

(b)

(c)

Figure 4.13: 1-1/2 D_{ips} spectrum of three complex sinusoids at an input SNR of -18dB. The representation is formed using a 128 point data window.

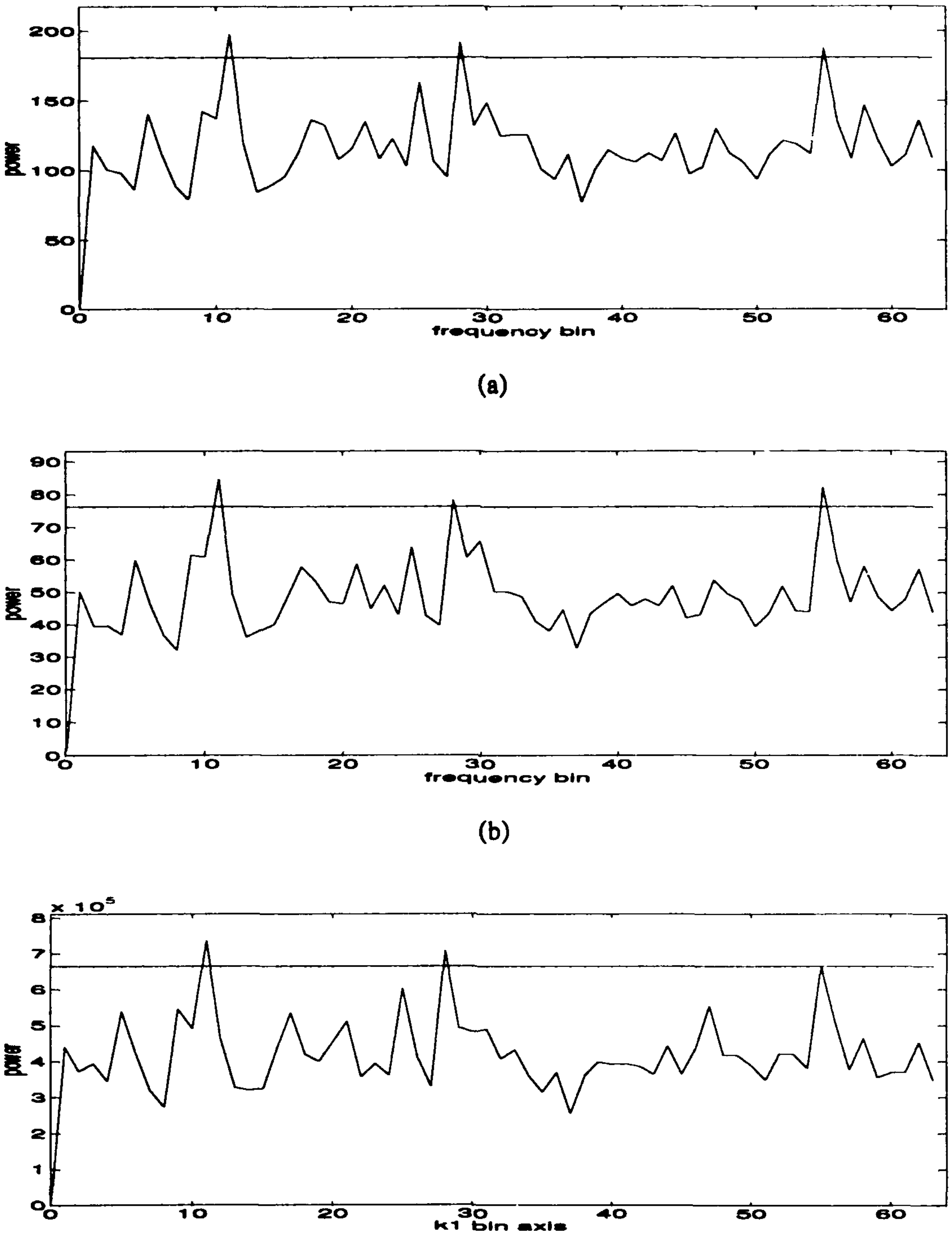


(a)

(b)

(c)

Figure 4.14: Bispectrum of three complex sinusoids at an input SNR of -18dB. The representation is formed using a 128 point data window.

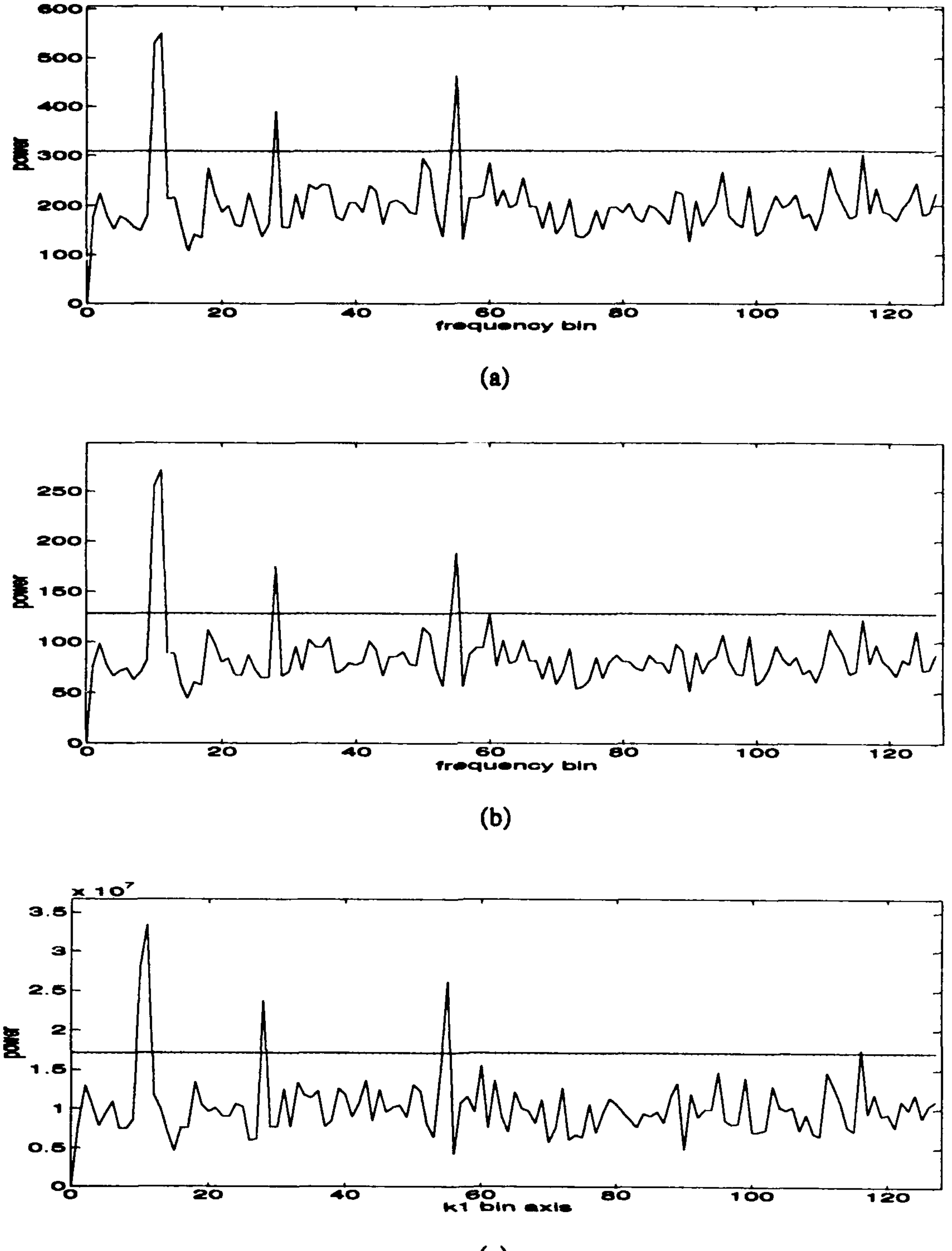


(a)

(b)

(c)

Figure 4.15: Frequency bin averages for an input SNR of -21dB and a 128 point window length; (a) spectrogram, (b) 1-1/2 D_{ips}, and (c) bispectrum.

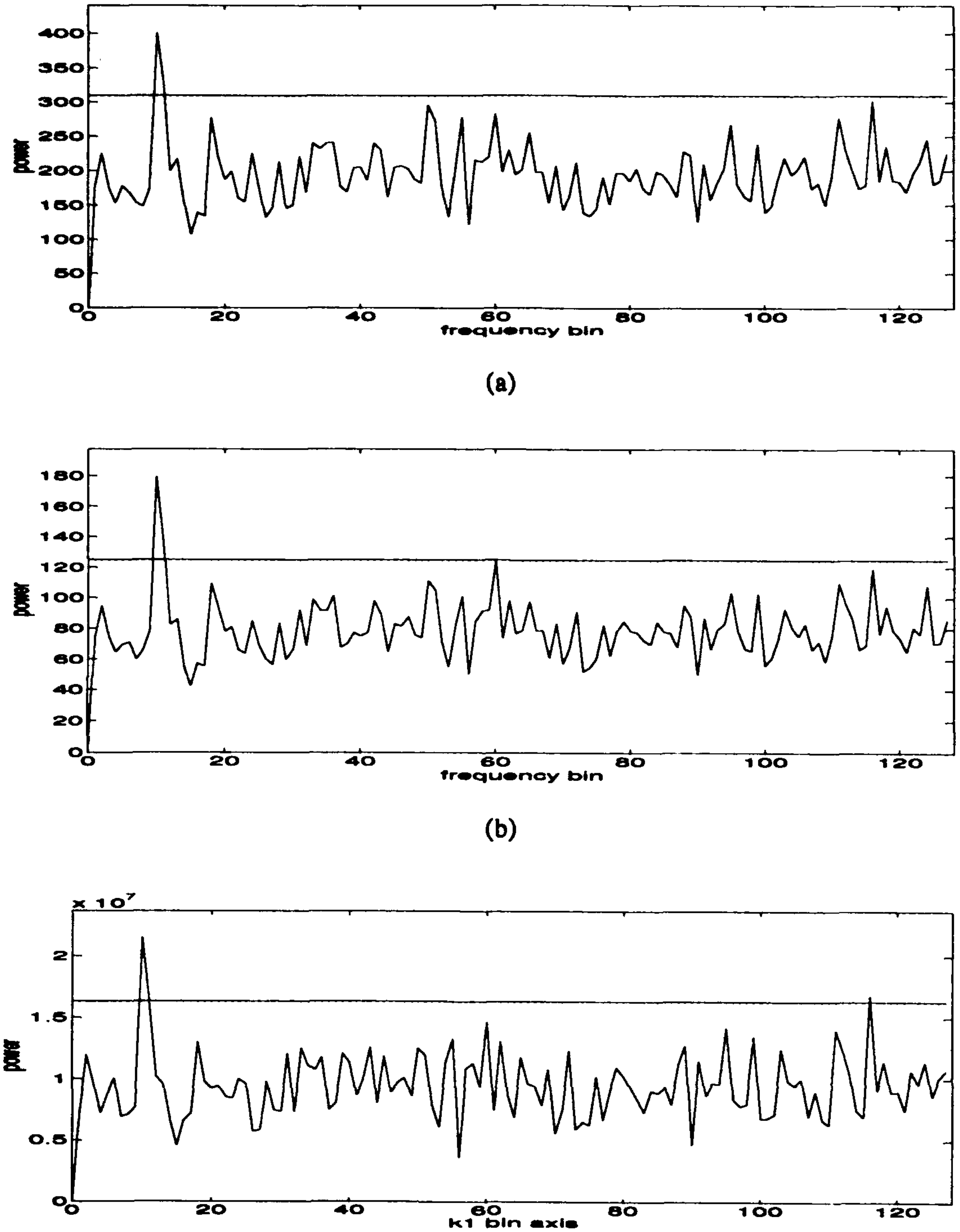


(a)

(b)

(c)

Figure 4.16: Frequency bin averages for an input SNR cf -21dB and a 256 point window length; (a) spectrogram, (b) 1-1/2 D_{ips}, and (c) bispectrum.



(a)

(b)

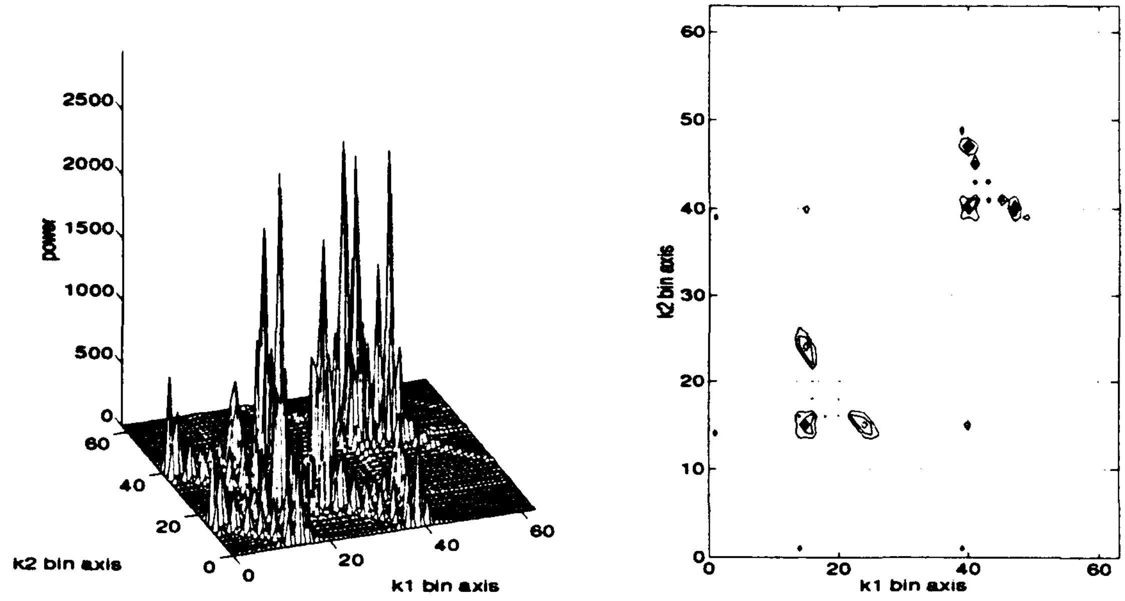
(c)

Figure 4.17: Frequency bin averages for an input SNR of -26dB and a 256 point window length; (a) spectrogram, (b) 1-1/2 D_{ips}, and (c) bispectrum.

B. HARMONICALLY RELATED SINUSOIDS

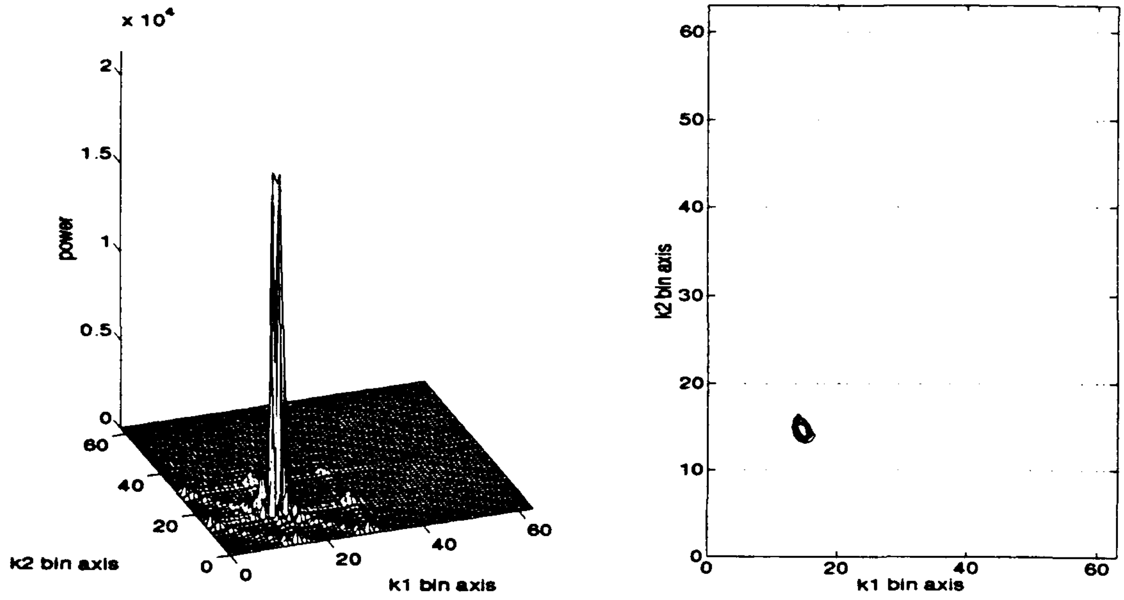
One of the main advantages of higher than second order spectra lies in their ability to preserve true phase information [Ref. 1]. A common application found in the literature to test this ability is the detection of quadratic phase coupling [Refs. 1,3,4,11]. Three sinusoids are quadratically phase coupled when they are both harmonically related and phase related. Harmonically related means that one of the sinusoids has a frequency equal to the sum of the other two. Similarly, phase related means that one of the phases is equal to the sum of the other two. [Ref. 1]

Simulations involving BISPEC_D show that the bispectrum is sensitive to signals containing harmonic components in general and not just when quadratic phase coupling occurs. A signal consisting of two sinusoids is considered initially. The first sinusoid has a frequency corresponding to bin 15, the second sinusoid's frequency corresponds to bin 40. Both sinusoids have uniform random phases independently distributed over the interval $(0, 2\pi)$. The bispectrum representation is generated using BISPEC_D with a 3^2 Rao-Gabr window, a 64 point data window, and 128 point FFTs. Figure 4.18 shows the bispectrum representation of this signal. Both components of the signal are indicated by the vertically aligned pair of peaks in k_1 bins 15 and 40, and/or by the horizontally aligned peaks in k_2 bins 15 and 40. Figure 4.19 displays the results for the same simulation when the second sinusoid's frequency is changed so that it now corresponds to bin 30 and becomes a harmonic of the first component. Only the one peak at $k_1 = k_2 = 15$ actually stands out on the bispectrum surface. The bispectrum is blind to the sinusoid at bin 30 because it is the highest harmonic component of the first sinusoid contained in the signal. A signal comprised of three harmonic components located at bins 15, 30, and 45 is now processed under two phase related conditions. For the first condition, all the phases are i.i.d. uniformly over $(0, 2\pi)$. The bispectrum representation for this signal is shown in



(a) (b)

Figure 4.18: Two unrelated sinusoids, one at bin 15 and the other at bin 40.



(a) (b)

Figure 4.19: Two harmonic sinusoids, one at bin 15 and the other at bin 30.

Figure 4.20. The highest harmonic component is again not represented by a significant peak located in bin 45. However, the bispectrum does indicate by its appearance that the signal contains harmonic components based on the fundamental frequency corresponding to bin 15. If the components were not harmonics of this fundamental frequency the bispectrum surface would have an appearance similar to that previously shown in Figure 4.11.

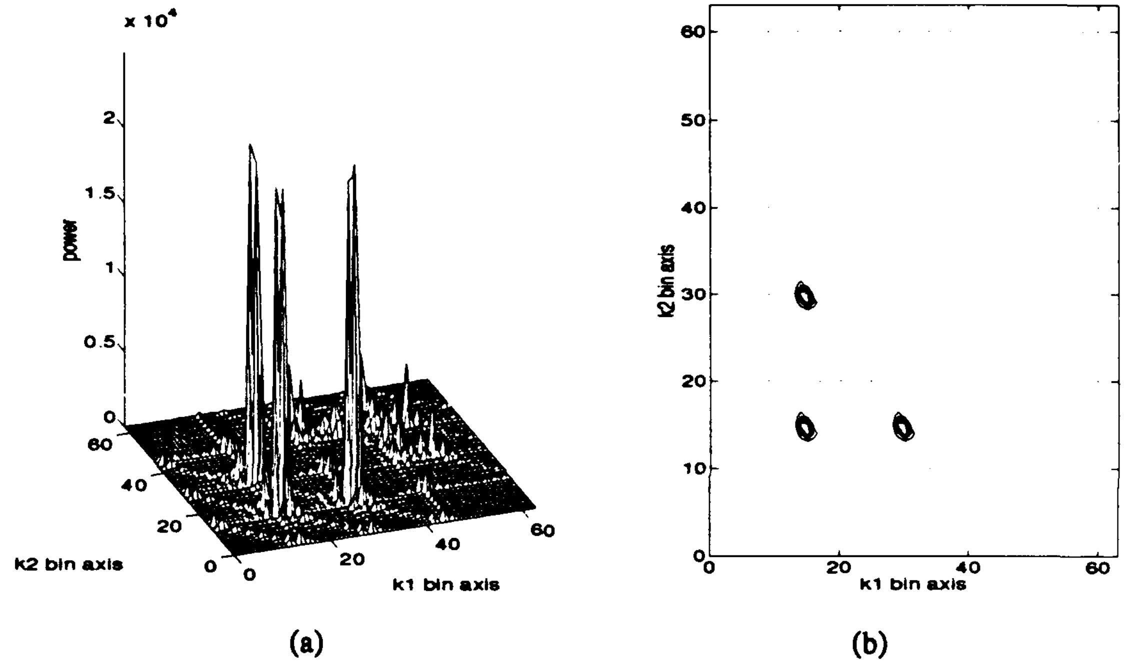
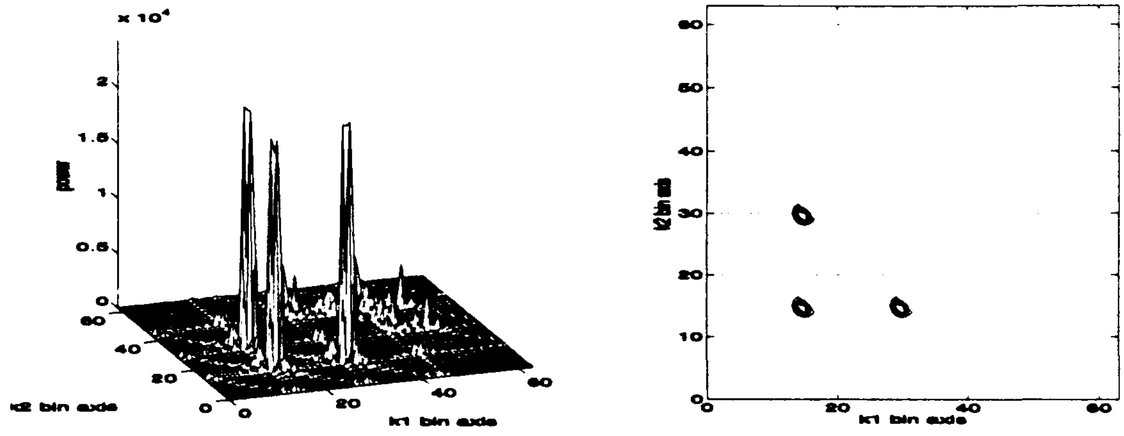


Figure 4.20: Three harmonic sinusoids in bins 15, 30, and 45 with unrelated phases.

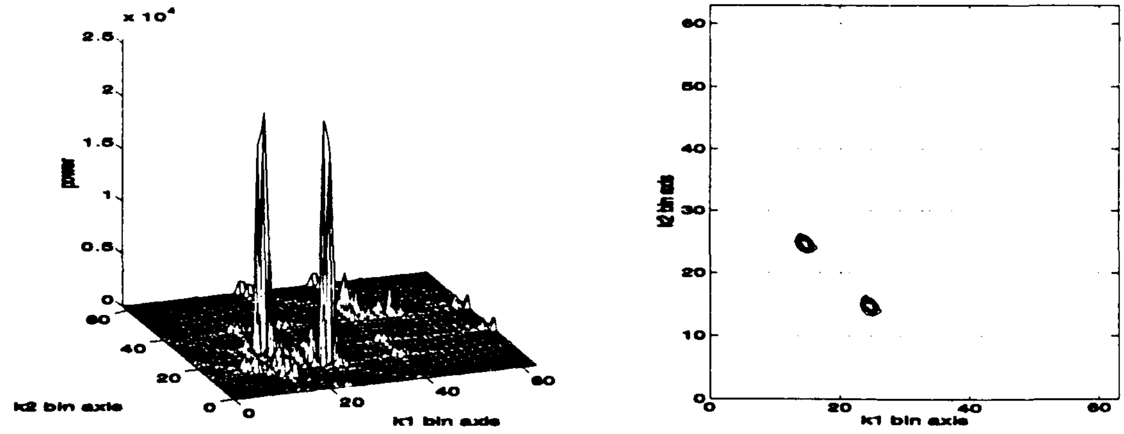
Figure 4.21 displays the bispectrum for the three harmonic component signal under the second phase condition. Here the third sinusoid has a phase equal to the sum of the other two component phases. This situation satisfies the definition of quadratic phase coupling. There is essentially no difference between this bispectrum and the one generated for the unrelated phase condition. When the signal components are harmonics of the same fundamental frequency, the bispectrum representation does not discern quadratic phase coupling.



(a) (b)

Figure 4.21: Three quadratically phase coupled harmonic sinusoids in bins 15, 30, and 45.

If the signal components are harmonically related but are not harmonics of the same fundamental frequency, the bispectrum does distinguish quadratic phase coupling. This frequently considered situation in the bispectrum literature is depicted in Figure 4.22. The signal components are quadratically coupled but the signal frequencies are 15, 25, and 40 rather than harmonics (i.e., 15, 30, and 45). This representation has just two peaks located at the intersection of the k_1 and k_2 bins where the first two sinusoids reside.



(a) (b)

Figure 4.22: Quadratically phase coupled sinusoids located in bins 15, 25, and 45.

C. MULTI-COMPONENT LINEAR CHIRPS

A computer simulation based upon Example 1 of [Ref. 6] is used to compare the processing ability of IHOMS, 1-1/2 D_{ips}, and the spectrogram when the received signal consists of two linear FM signals mixed with white Gaussian noise. In terms of the variables in (2.27),

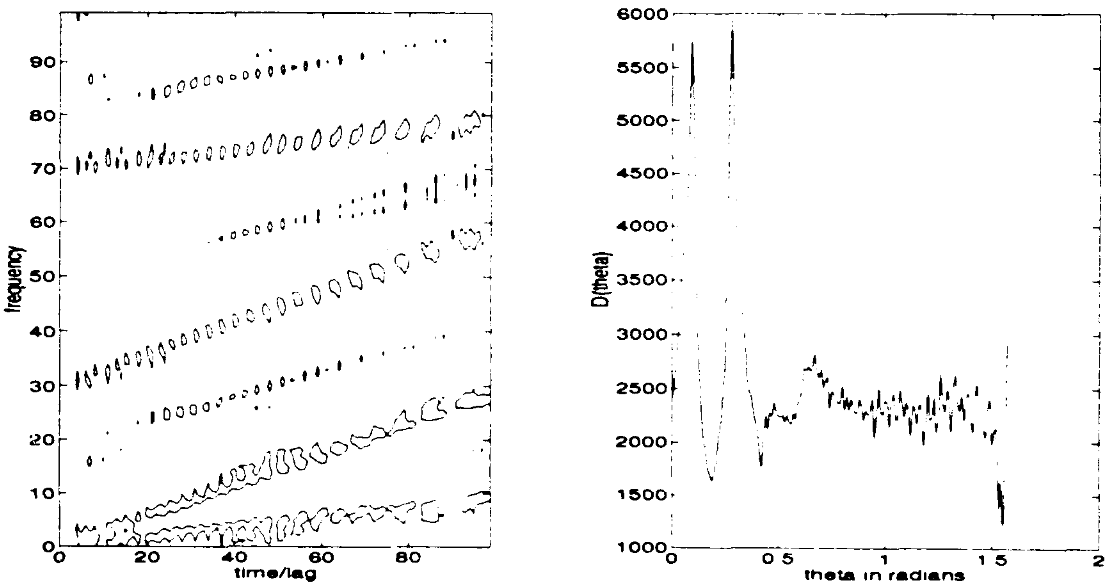
$$x(n) = \sum_{i=1}^{M} exp\{j2\pi\left[(\frac{n}{N})u_i + (\frac{n}{N})^2 b_i\right]\},$$

M is two, $u_1 = 8$, $b_1 = 94$, $u_2 = 196$, $b_2 = 30$, and $N = 256$. With a record length equal to one second and a sampling rate of 256 samples/second, chirp 1 exhibits a slew rate of 188 Hz/second, starts at a frequency of 8/256, and finishes at 196/256. Likewise, chirp 2 has a slew rate of 60 Hz/second, starts at a frequency of 196/256, and finishes at a frequency equal to 256/256. The parameter u_i is the fractional starting frequency of the i^{th} chirp times N. The parameter b_i is equal to half of the fractional frequency difference of the i^{th} chirp times N, or simply half of the i^{th} chirp's slew rate.

The IHOMS method was applied first by following the procedure outlined on pages 19 through 21. Choosing to work with a fourth order moment slice of the signal fixes the length of the parameter vector, a, to four (i.e., $\underline{a} = [1, a_1, a_2, a_3]$). GEN_A is an extrinsic Matlab function in Appendix C used to generate a desired number of different parameter vectors having elements that satisfy (2.28) and (2.29). Three parameter vectors are formed for the first simulation. The remaining steps in the procedure are carried out by the extrinsic Matlab function ATH_IMS in Appendix C. One-hundred lags are used to compute the fourth-order moment slice by (2.30) and one time-frequency representation is calculated in accordance with (2.31) for each of the three parameter vectors. The value for the parameter g in the exponent of the kernel function in (2.31) is set equal to 0.0008 because the same value was found to give acceptable results in Example 1 of [Ref. 6]. The final time-frequency representation is obtained by computing the magnitude of the three parameter vector specific surfaces. Figure 4.23(a) shows an IHOMS time-frequency representation based upon a fourth order moment slice and three parameter vectors for a noise-free signal consisting of chirp 1 and chirp 2. The two chirps are indicated by the two contour lines radiating from the origin. The remaining lines on the plot are cross-terms. Figure 4.23(b) is a plot of the one-dimensional function $D(\theta)$ given by (2.33). This function represents the magnitude of the time-frequency surface along radial lines passing through the origin of the surface. The first maximum in Figure 4.23(b) occurs at 0.09 radians and corresponds to the b parameter of chirp 2 through (2.34):

$$\hat{b}_2 = \frac{256^2}{2(100)} \tan(0.09) = 29.57.$$

Applying (2.34) to the location of the second maximum, 0.28 radians, an estimate of the b parameter of the first chirp, $\hat{b}_1 = 94.23$, is obtained.



(a) (b)

Figure 4.23: Three parameter vector IHOMS representation for a noise-free signal.

Figure 4.24 shows the three parameter vector IHOMS representation for the same two chirps mixed with white Gaussian noise at an input SNR of -3dB. The chirps cannot be distinguished from the background noise.

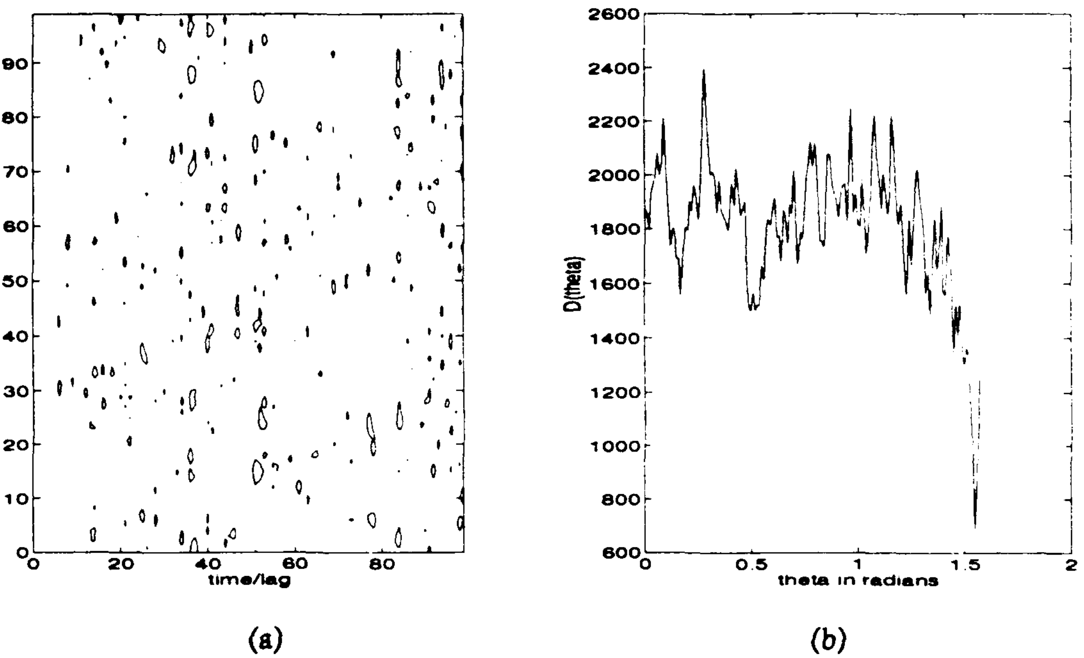
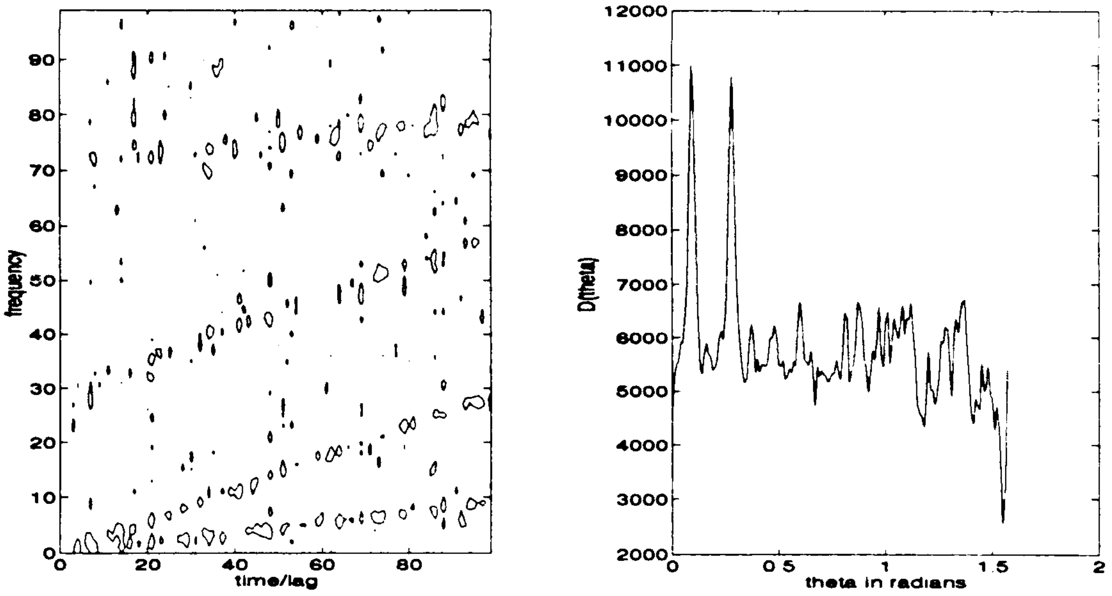


Figure 4.24: Three parameter vector IHOMS representation when SNR = -3dB.

Signal detection is improved by summing more parameter vector specific surfaces in order to form the IHOMS representation. This is accomplished by creating more parameter vectors. The IHOMS representation in Figure 4.25 uses 21 parameter vectors on the two chirp signal at an SNR of -3dB. The chirps are once again discernible. Even with 21 parameter vectors the detection capability degrades significantly as the SNR becomes worse. Whether the chirps can still be distinguished in Figure 4.26 when the SNR is -5.5dB is questionable. At lower SNRs they definitely become lost in the noise.

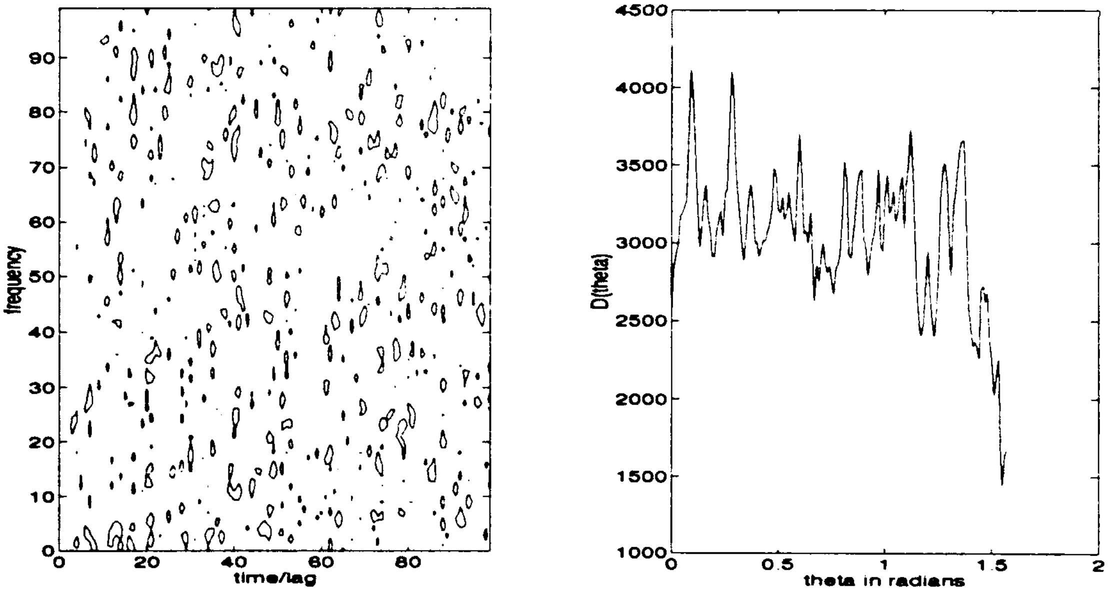
The 1-1/2D$_{ips}$ [illegible] spectrogram estimates of the double chirp signal are obtained using ONE_HALF and SPECTRO respectively. A rectangular window function, a 32 point



(a)

(b)

Figure 4.25: Twenty-one parameter vector IHOMS representation when SNR = -3dB.

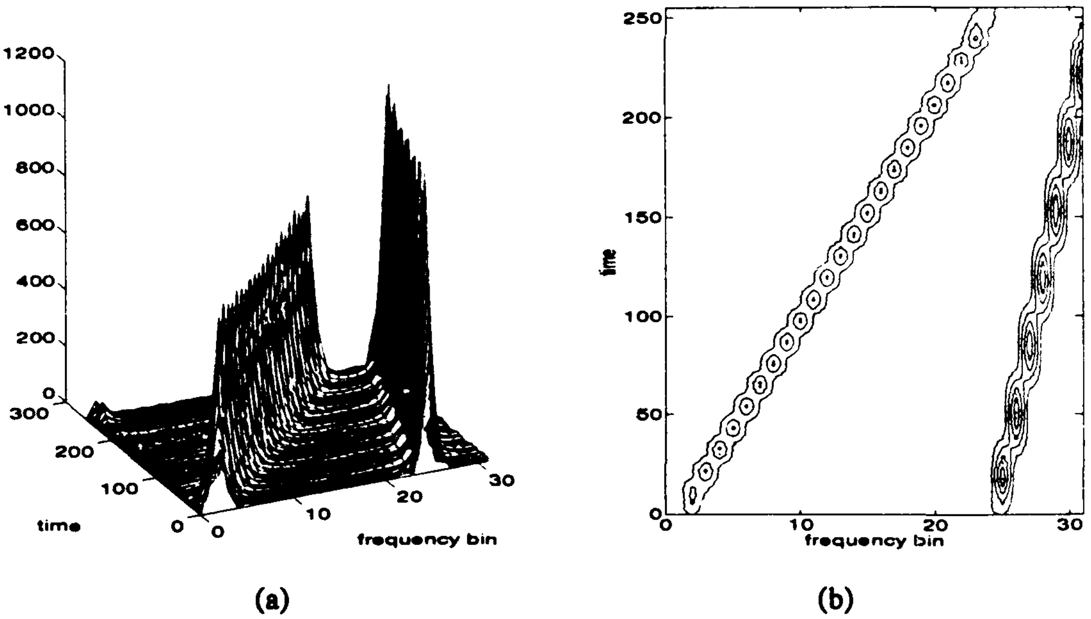


(a)

(b)

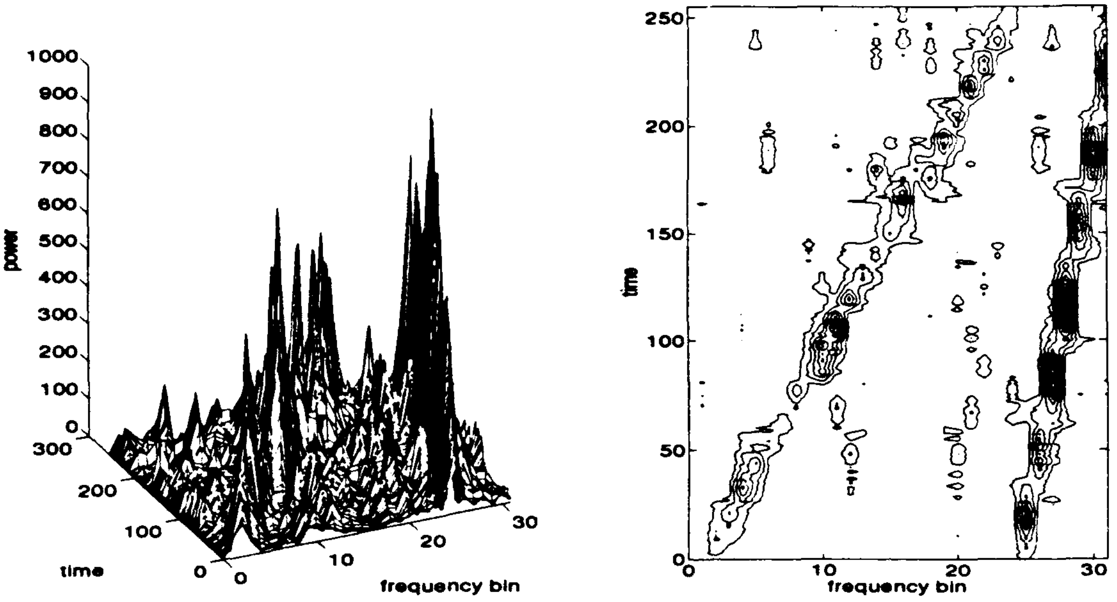
Figure 4.26: Twenty-one parameter vector IHOMS representation when SNR = -5.5dB.

data window, and a step increment of one are the specified function options used in the simulations. Figures 4.27, 4.28, and 4.29 show the spectrogram time-frequency representations for the noise-free signal, an SNR of -3dB, and an SNR of -5.5dB, respectively. Figures 4.30, 4.31, and 4.32 display the 1-1/2 D_{ips} representations for the noise-free case, an SNR of -3dB, and an SNR of -5.5dB, respectively. Unlike the IHOMS representation, both 1-1/2 D_{ips} and the spectrogram indicate fractional start and finish frequencies for each chirp in addition to slew rate. For a given noise level, the detection capabilities of 1-1/2 D_{ips}, the spectrogram, and the 21 parameter vector IHOMS are comparable. However, IHOMS has a much higher computational cost than either of the other methods. Generating a 21 parameter vector IHOMS surface based upon a moment slice calculated from 100 lags requires 2,100 100-point Fourier transforms. By comparison, the spectrogram and 1-1/2 D_{ips} require only 256 32-point Fourier transforms to generate an equivalent time-frequency representation.



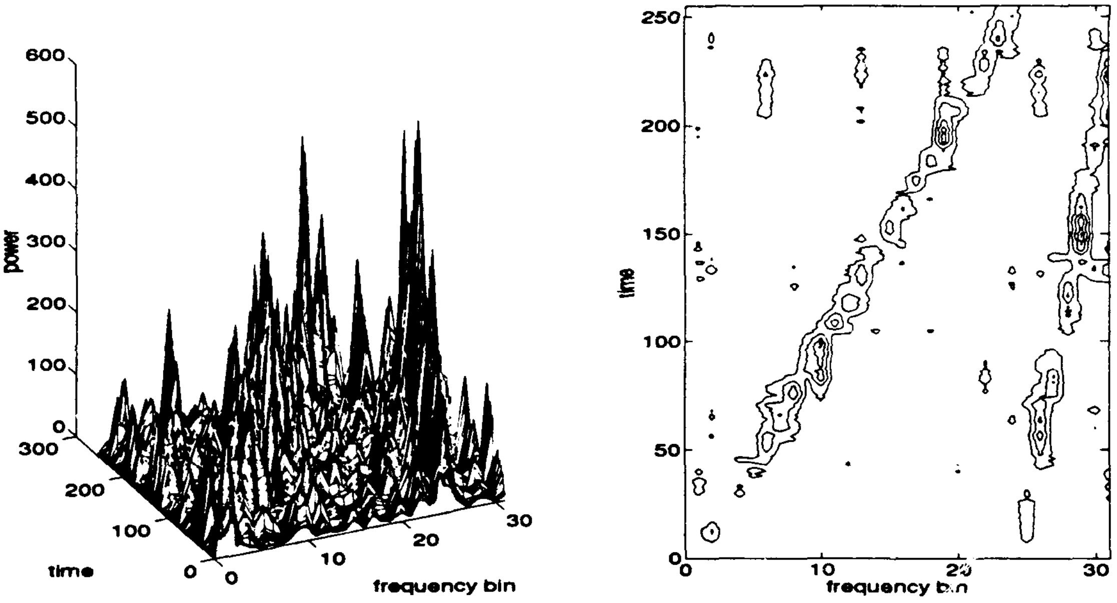
(a) (b)

Figure 4.27: Spectrogram of noise-free two chirp signal.



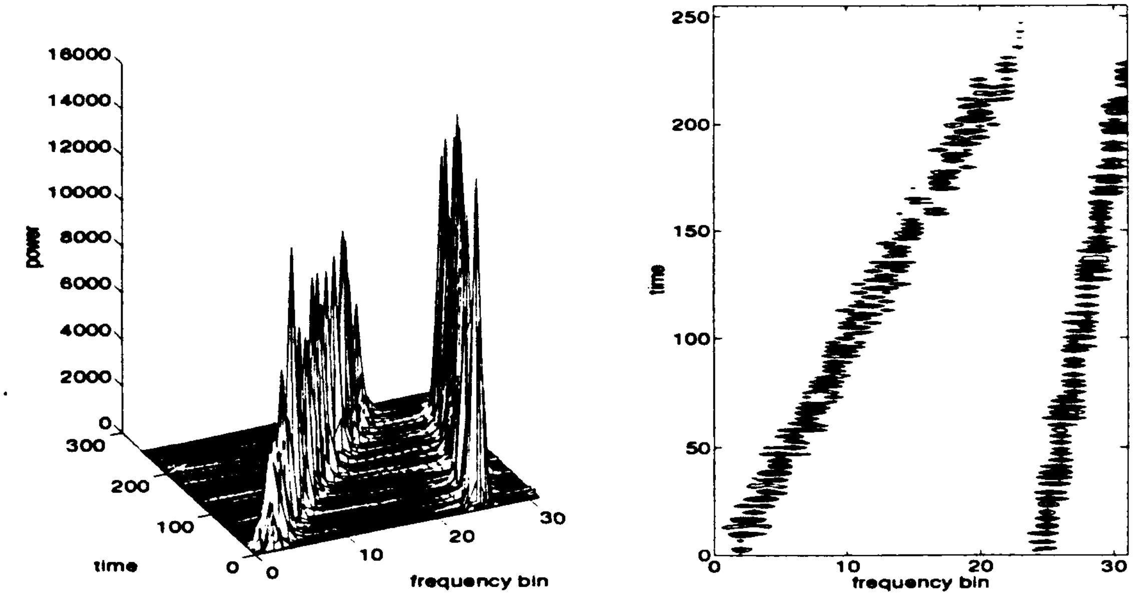
(a) (b)

Figure 4.28: Spectrogram of two chirp signal with SNR = -3dB.



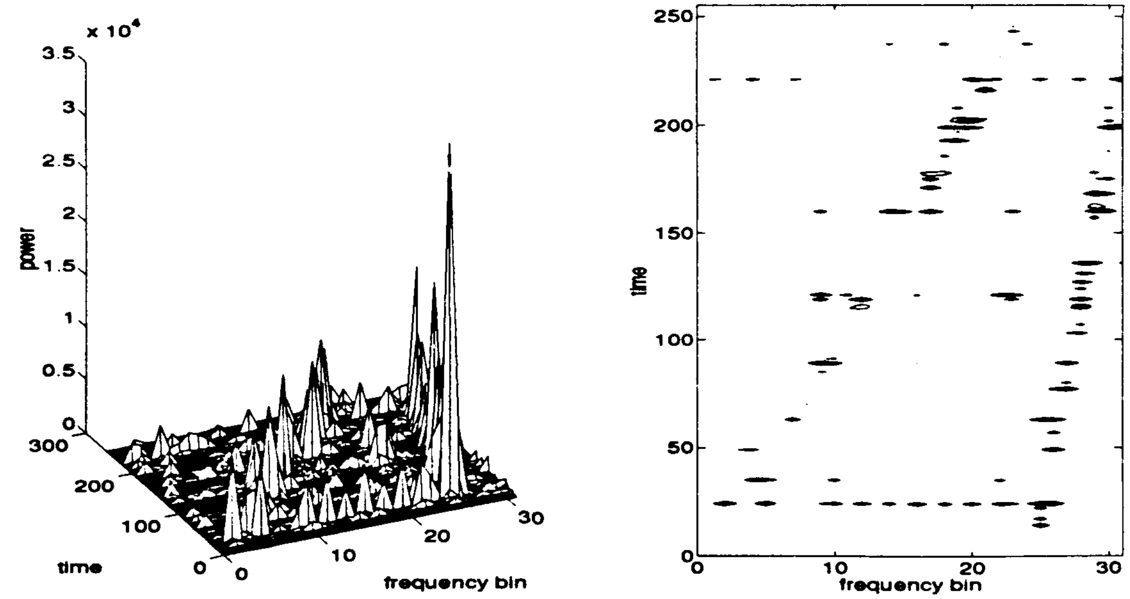
(a) (b)

Figure 4.29: Spectrogram of two chirp signal with SNR = -5.5dB.



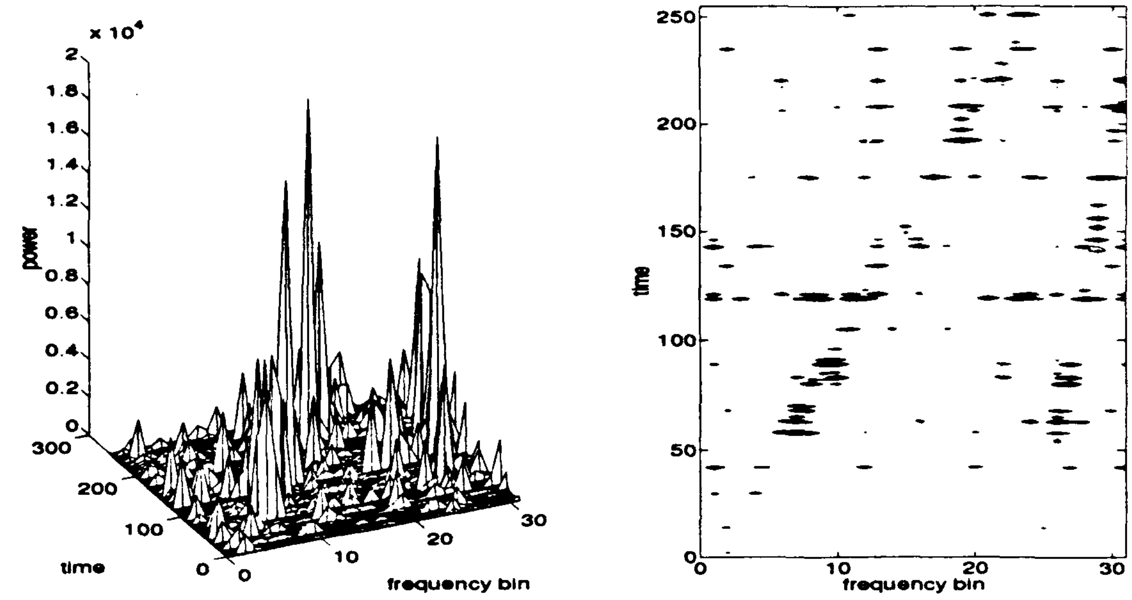
(a) (b)

Figure 4.30: 1-1/2 D_{ips} representation of noise-free two chirp signal.



(a) (b)

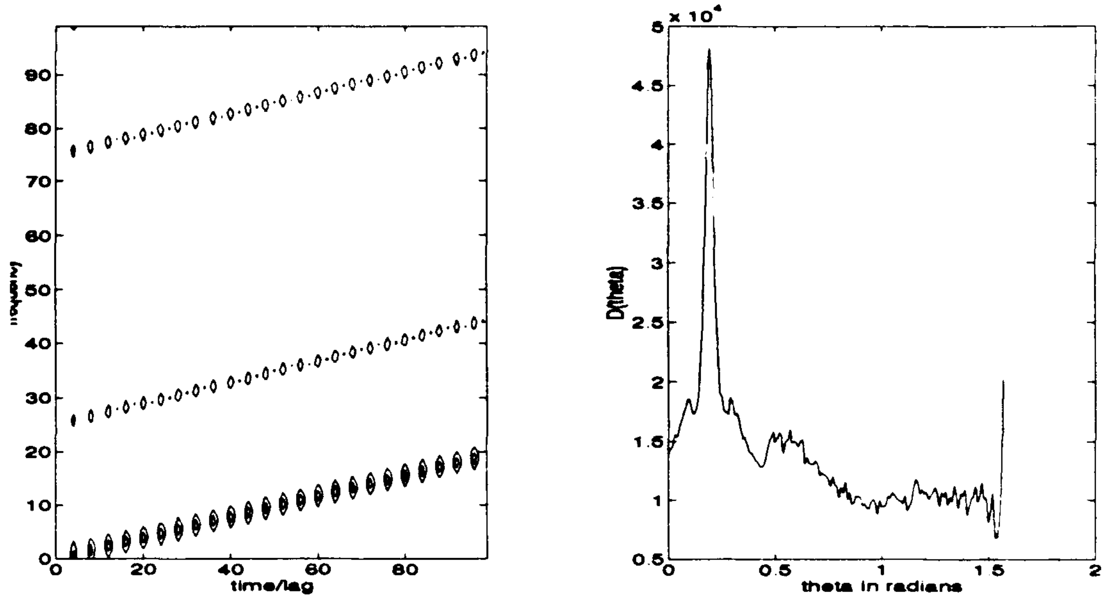
Figure 4.31: 1-1/2 D_{ips} representation of two chirp signal with SNR = -3dB.



(a) (b)

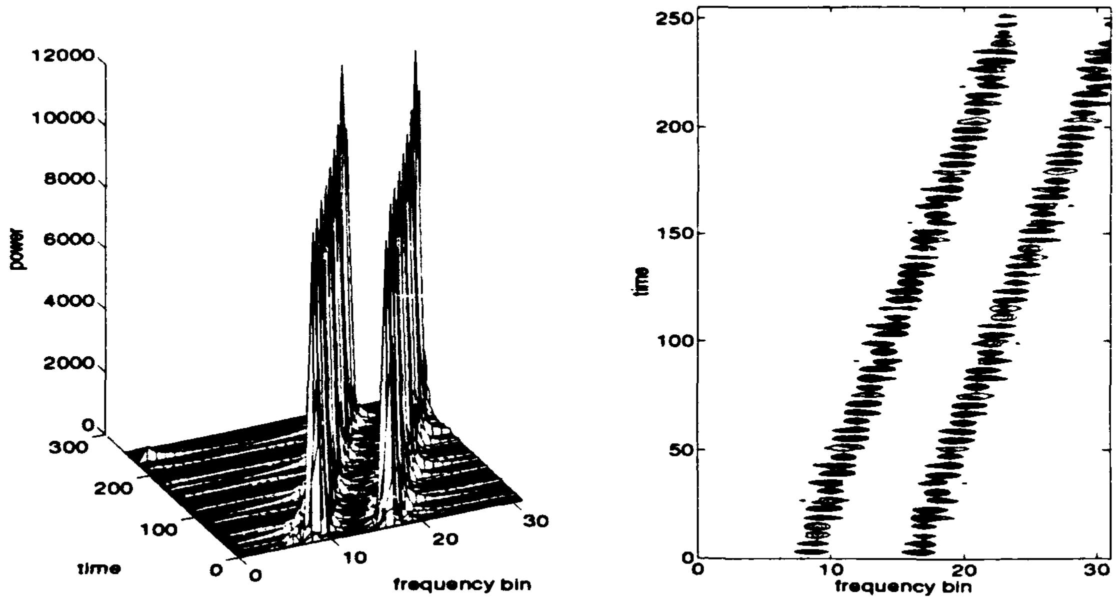
Figure 4.32: 1-1/2 D_{ips} representation of two chirp signal with SNR = -5.5dB.

IHOMS representations indicate only the slew rate of chirps and not their start/stop frequencies. As a consequence, IHOMS cannot distinguish between two or more different chirps having the same slew rate. Figure 4.33 displays the IHOMS representation for two chirps. The first chirp starts at a frequency of 64/256 and finishes at 192/256. The second chirp starts at 128/256 and finishes at 256/256. They both have the same slew rate of 128Hz/second. The IHOMS representation detects the presence of only one chirp in the signal. The maximum of $D(\theta)$ occurs at 0.19 radians which corresponds to the proper b parameter of 64. The spectrogram and 1-1/2 D_{ips} representations indicate the presence of both chirps. The 1-1/2 D_{ips} time-frequency surface of this signal is displayed in Figure 4.34.



(a) (b)

Figure 4.33: IHOMS representation of two chirps having the same slew rate.



(a) (b)

Figure 4.34: 1-1/2 D_{ips} representation of two chirps having the same slew rate.

The IHOMS algorithm also requires a more intricate sampling scheme than does the spectrogram and 1-1/2 D_{ips} methods. The received signal has to be sampled at a much higher sampling rate in order to obtain the data samples needed to form the moment slices in accordance with (2.30) while satisfying (2.28) and (2.29).

V. CONCLUSIONS

A. DISCUSSION OF RESULTS

The 1-1/2 D_{ips} method handles both signal and noise power differently than the spectrogram. With respect to signal power, the maximum processed signal power is the same for either method. But, because 1-1/2 D_{ips} signal power is modulated, the average signal power is either a third of the spectrogram's if the signal is real valued, or half the spectrogram's if the signal is complex valued. With respect to noise power, the methods exhibit estimate variances that depend differently on the size of the data window and the input variance. Whereas the spectrogram's variance is a function of input variance squared and window length squared, 1-1/2 D_{ips}'s variance depends upon input variance cubed and window length. Consequently, 1-1/2 D_{ips} is more resistant to input noise than the spectrogram if the noise level is low enough and the window length is large enough. Conversely, the spectrogram handles noise better if either the window is too small or the input noise is too great. When the effects of signal and noise are combined, the methods are compared by determining how many noise standard deviations the average signal power is above the mean noise level. This measure represents the method's processing gain. 1-1/2 D_{ips} has a larger processing gain than the spectrogram at high SNR for any reasonably sized data window. As SNR decreases, the processing gain advantage enjoyed by 1-1/2 D_{ips} diminishes. At a sufficiently low SNR value that depends on the size of the data window, the spectrogram becomes the better estimate. As window length increases, 1-1/2 D_{ips}'s processing gain improves relative to the spectrogram's.

As implemented in this study, the spectrogram and 1-1/2 D_{ips} are superior to the bispectrum at detecting multi-component stationary signals in white Gaussian noise. However, the frequency bin averaging scheme used in the simulations is not as favorable to the bispectrum due to the nature of its frequency-frequency representation. The bispectrum does have an advantage though over the spectrogram and 1-1/2 D_{ips} in its ability to distinguish between unrelated signals, harmonic signals, and quadratically phase coupled signals. The only limitation is that quadratic phase coupling is not detectable if the frequencies of the coupled signals are harmonics of the same fundamental frequency.

IHOMS usefulness is limited to detection of linear chirps having different slew rates. This method is considerably more costly than the spectrogram and 1-1/2 D_{ips} in terms of computations, especially for noisy signals.

B. SUGGESTIONS FOR FUTURE STUDY

Future work on two aspects of 1-1/2 D_{ips} might prove worthwhile. The first is finding an optimum window and/or smoothing function. Rectangular windows, Hamming windows, and boxcar smoothing have been used in ONE_HALF to date. Since 1-1/2 D_{ips} is a degenerate form of the bispectrum, a window function satisfying bispectral window requirements might prove to be better window. The second area warranting further study involves the inherent modulating effect of the 1-1/2 D_{ips} algorithm. The modulation contains useful signal information if it could be extracted from the 1-1/2 D_{ips} representation.

The logical next step for future work on the bispectrum is to extend it to a time-frequency representation. Implementing an alternate bispectral calculation method relying on polar rasters looks like a promising starting point. Evaluation of parametric methods and comparison to the non-parametric methods of computing the bispectrum might prove worthwhile. In addition, better averaging schemes for the bispectrum frequency-frequency surface could be developed.

APPENDIX A: VARIANCE OF THE 1-1/2 D$_{\text{ips}}$ ESTIMATE

The 1-1/2 D$_{\text{ips}}$ spectrum is computed using (2.24). The variance of this estimate is derived here for a zero mean i.i.d. Gaussian input, $x \sim N(0, \sigma_x^2)$. Results are obtained for both real and complex valued noise. A rectangular window, i.e. $w(k)$=1 for $0 \leq k \leq N-1$, is assumed for these calculations.

The mean of the 1-1/2 D$_{\text{ips}}$ estimate is:

$$E\{1\text{-}1/2\ \hat{D}_{ips}\} = \frac{1}{2}\sum_{k=0}^{N-1} E\{|x(n)|^2 x^*(n-k) + |x(n)|^2 x(n+k)\}\, exp(-j\omega k) .$$

Since the expectation operation is distributive,

$$E\{1\text{-}1/2\ \hat{D}_{ips}\} = \frac{1}{2}\sum_{k=0}^{N-1} \{E[|x(n)|^2 x^*(n-k)] + E[|x(n)|^2 x(n+k)]\}\, exp(-j\omega k) .$$

Let $term1 = E\{|x(n)|^2 x^*(n-k)\}$ and $term2 = E\{|x(n)|^2 x(n+k)\}$. Expanding $term1$ yields

$$term1 = E\{x^*(n)x(n)x^*(n-k)\} .$$

For $k \neq 0$,

$$\begin{aligned} term1 &= E\{x^*(n)x(n)\}E\{x^*(n-k)\} \\ &= \sigma_x^2\mu_x \\ &= 0, \end{aligned}$$

since the input samples are independent and the mean of the input sequence, $\mu_x = 0$.

For $k = 0$,

$$term1 = E\{x^*(n)x(n)x^*(n)\}.$$

Assuming that the real and imaginary parts of $x(n)$ are independent,

$$x_r(n), x_i(n) \sim N(0, \frac{1}{2}\sigma_x^2),$$

then *term*1 can be rewritten as:

$$term1 = E\{ [x_r(n) - jx_i(n)]\ [x_r(n) + jx_i(n)]\ [x_r(n) - jx_i(n)] \},$$

which simplifies to,

$$term1 = \{E[x_r^3(n)] + E[x_r(n)]E[x_i^2(n)]\} - j\{E[x_i^3(n)] + E[x_i(n)]E[x_r^2(n)]\}$$

after cross-multiplying and regrouping the real and imaginary parts. The terms in the last expression involving the product of a first moment and a second moment are zero because the first moment is zero. The third moment quantities in the same expression are also zero because the processes are distributed normally about a zero-mean. The final result is that *term*1 is zero for any value of *k*. The same is true for *term*2. Consequently,

$$\begin{aligned} E\{1\text{-}1/2\ \hat{D}_{ips}\} &= \frac{1}{2}\sum_{k=0}^{N-1} \{term1 + term2\}\, exp(-j\omega k) \\ &= 0. \end{aligned}$$

The second moment of the estimate is:

$$E\{|1\text{-}1/2\ \hat{D}_{ips}|^2\} = E\{1\text{-}1/2\ \hat{D}_{ips}^{(m)}(n,\omega) \cdot 1\text{-}1/2\ \hat{D}_{ips}^{(s)*}(n,\omega)\}, \qquad (A.1)$$

where,

$$1\text{-}1/2\ \hat{D}_{ips}^{(m)}(n,\omega) = \frac{1}{2}\sum_{m=0}^{N-1} \{|x(n)|^2 x^*(n-m) + |x(n)|^2 x(n+m)\}\, exp(-j\omega m),$$

and

$$1\text{-}1/2\ \hat{D}_{ips}^{(s)*}(n,\omega) = \frac{1}{2}\sum_{s=0}^{N-1} \{|x(n)|^2 x(n-s) + |x(n)|^2 x^*(n+s)\}\, exp(j\omega s).$$

After multiplying the last two equations together, collecting terms, and distributing the expectation operator, (A.1) becomes:

$$E\{|1\text{-}1/2\ \hat{D}_{ips}|^2\} = \frac{1}{4}\sum_{m=0}^{N-1}\sum_{s=0}^{N-1}\{term1 + term2 + term3 + term4\}\, exp\,(-j\omega\,(m-s)) \qquad \text{(A.2)}$$

where $term1 = E\,[|x(n)|^4 x^*(n-m)\,x(n-s)]\,,$

$term2 = E\,[|x(n)|^4 x^*(n-m)\,x^*(n+s)]$

$term3 = E\,[|x(n)|^4 x(n+m)\,x(n-s)]\,,$

and $term4 = E\,[|x(n)|^4 x(n+m)\,x^*(n+s)]\,.$

The first element in each of the terms above represents a product of four elements:

$$|x(n)|^4 = x^*(n)\,x(n)\,x^*(n)\,x(n)\,.$$

*Term*1, through *term*4 are considered separately for all possible m and s index values. Two different mixed-moment relationships for a zero-mean Gaussian process are needed. For a real, zero mean Gaussian process, the following relation applies [Ref. 7]:

$$\begin{aligned} E\{x_1x_2x_3x_4\} &= E\{x_1x_2\}\,E\{x_3x_4\} + E\{x_1x_3\}\,E\{x_2x_4\} \\ &\quad + E\{x_1x_4\}\,E\{x_2x_3\}\,. \end{aligned} \qquad \text{(A.3)}$$

For a complex zero mean Gaussian process, a slightly different relation holds [Ref. 11]:

$$E\{x^*{}_1x_2x^*{}_3x_4\} = E\{x^*{}_1x_2\}\,E\{x^*{}_3x_4\} + E\{x^*{}_1x_4\}\,E\{x^*{}_3x_2\}\,. \qquad \text{(A.4)}$$

*Term*1 is evaluated first:

1. For $m \neq 0$ and $s \neq 0$,

$$term1 = E\,[x^*(n)\,x(n)\,x^*(n)\,x(n)]\,E\,[x^*(n-m)\,x(n-s)]\,. \qquad \text{(A.5)}$$

Applying (A.3) to a real process, *term*1 reduces to:

$$\begin{aligned} term1^R &= 3\,[E\{x(n)\,x(n)\}]^2 E\{x(n-m)\,x(n-s)\} && \text{(since i.i.d.)} \\ &= 3\,(\sigma_x^2)^2 R_x(s-m) && \text{(since zero-mean)} \end{aligned}$$

$$= 3\sigma_x^4 \cdot \sigma_x^2 \delta(s-m) \qquad \text{(since i.i.d.)}$$

$$= 3\sigma_x^6 \delta(s-m). \tag{A.6}$$

For a complex process, (A.4) is employed on (A.5) to achieve a similar result:

$$\begin{aligned} term1^C &= 2[E\{x(n)x^*(n)\}]^2 E\{x^*(n-m)x(n-s)\} \\ &= 2(\sigma_x^2)^2 R_x(m-s) \\ &= 2\sigma_x^4 \cdot \sigma_x^2 \delta(m-s) \\ &= 2\sigma_x^6 \delta(m-s). \end{aligned} \tag{A.7}$$

2. If $m = 0$ and $s \neq 0$,

$$term1^C = E\{x^*(n)x(n)x^*(n)x(n)x^*(n)\}E\{x(n-s)\} \tag{A.8}$$

$$= 0 \quad (\text{zero-mean process} \Rightarrow E\{x(n-s)\} = 0)$$

$$= term1^R.$$

This result is the same for both a real process and a complex process.

3. If $m \neq 0$ and $s = 0$,

$$term1^C = E\{x^*(n)x(n)x^*(n)x(n)x(n)\}E\{x^*(n-m)\} \tag{A.9}$$

$$= 0 \quad (\text{zero-mean process} \Rightarrow E\{x^*(n-m)\} = 0)$$

$$= term1^R.$$

4. If $m = s = 0$,

$$term1^C = E\{x^*(n)x(n)x^*(n)x(n)x^*(n)x(n)\}. \tag{A.10}$$

Assuming that the real and imaginary parts of the process are independent implies that,

$$x_r(n), x_i(n) \sim N(0, \frac{1}{2}\sigma_x^2) \tag{A.11}$$

Dropping the index notation in (A.10), breaking up each term into its real and imaginary parts, and grouping like terms produces:

$$term1^C = E\{(x_r - jx_i)^3 (x_r + jx_i)^3\},$$

which, after multiplication and collection of terms, becomes:

$$term1^C = E\{x_r^6\} + 3E\{x_r^2\}E\{x_i^4\} + 3E\{x_r^4\}E\{x_i^2\} + E\{x_i^6\}. \qquad (A.12)$$

The second order moments in (A.12) are simply the variance of the real and imaginary parts of the process. The fourth order moments are computed using (A.3). The sixth order moments are obtained by computing the sixth derivative of the moment generating function for a zero-mean Normal process [Ref. 17]:

$$\begin{aligned} E\{p^6\} &= \Phi_p^{(6)}(t)\Big|_{t=0} = \frac{d^6}{dt^6}exp\left(\frac{\sigma_p^2 t^2}{2}\right)\Bigg|_{t=0} \\ &= 15\sigma_p^6; \end{aligned}$$

where the subscript p denotes either x_r or x_i since their variances are equal. For a complex process, in terms of this variance, (A.12) becomes:

$$\begin{aligned} term1^C &= 15\sigma_p^6 + 3(\sigma_p^2)(3\sigma_p^4) + 3(3\sigma_p^4)(\sigma_p^2) + 15\sigma_p^6 \\ &= 48\sigma_p^6. \end{aligned} \qquad (A.13)$$

From (A.11), $\sigma_p^2 = \frac{1}{2}\sigma_x^2$. Using this relation, (A.13) becomes:

$$\begin{aligned} term1^C &= 48\left[\frac{\sigma_x^2}{2}\right]^3 \\ &= 6\sigma_x^6. \end{aligned} \qquad (A.14)$$

If the process is real, the x_i terms in (A.12) are zero and x_r terms become x:

$$term1^R = E\{x^6\}.$$

Evaluating the sixth moment using the moment generating function yields:

$$term1^R = 15\sigma_x^6. \qquad (A.15)$$

*Term*2 was similarly analyzed. For the $m \neq 0$ and $s \neq 0$ index combination, the second moment term in the equivalent (A.6) and (A.7) is:

$$\begin{aligned} E\{x^*(n-m)x(n+s)\} &= \sigma_x^2\delta(-m-s) \\ &= 0, \end{aligned}$$

because the delta function is always zero for non-negative indices. Like *term*1, *term*2 is zero whenever just one index is zero because the process is zero mean. If the process is real and both indices are zero then *term*2 becomes a sixth order moment:

$$\begin{aligned} term1^R &= E\{x(n)x(n)x(n)x(n)x(n)x(n)\} \\ &= 15\sigma_x^6. \end{aligned}$$

If the process is complex, four terms in the expectation are conjugated while two are not. The result is a final expected value of zero since the process is zero mean Gaussian with independent real and imaginary parts:

$$\begin{aligned} term1^C &= E\{x^*(n)x(n)x^*(n)x(n)x^*(n)x^*(n)\} \\ &= E\{(x_r - jx_i)^4(x_r + jx_i)^2\} \\ &= 0. \end{aligned}$$

Consequently, *term*2 is zero for all index combinations if the process is complex and is non-zero only for the $m = s = 0$ condition if the process is real.

Evaluation of *term*3 produces the same result obtained as for *term*2, and evaluation of *term*4 produces the same results obtained as for *term*1. Table A.1, and Table A.2 summarize term evaluation results for the real and complex processes, respectively.

TABLE A.1: SUMMARY OF TERMS FOR REAL PROCESSES

index condition	term1	term2	term3	term4
$m \neq 0$ and $s \neq 0$	$3\sigma_x^6\delta(s-m)$	0	0	$3\sigma_x^6\delta(s-m)$
$m = 0$ and $s \neq 0$	0	0	0	0
$m \neq 0$ and $s = 0$	0	0	0	0
$m = s = 0$	$15\sigma_x^6$	$15\sigma_x^6$	$15\sigma_x^6$	$15\sigma_x^6$

TABLE A.2: SUMMARY OF TERMS FOR COMPLEX PROCESSES

index condition	term1	term2	term3	term4
$m \neq 0$ and $s \neq 0$	$2\sigma_x^6\delta(s-m)$	0	0	$2\sigma_x^6\delta(s-m)$
$m = 0$ and $s \neq 0$	0	0	0	0
$m \neq 0$ and $s = 0$	0	0	0	0
$m = s = 0$	$6\sigma_x^6$	0	0	$6\sigma_x^6$

To obtain the second moment of the 1-1/2 D$_{\text{ips}}$ estimate for a real process, the terms shown in Table A.1 are substituted into (A.2). The $m = s = 0$ indices are taken out of the summation to obtain:

$$E\{|1\text{-}1/2\ \hat{D}_{ips}|^2\}_{REAL} = \frac{1}{4}\{4(15\sigma_x^6) + \sum_{m=1}^{N-1}\sum_{s=1}^{N-1} 2(3\sigma_x^6)\,\delta(s-m)\,exp(-j\omega(m-s))\}$$

$$= \frac{1}{4}\{60\sigma_x^6 + \sum_{m=1}^{N-1} 2(3\sigma_x^6)\,exp(-j\omega m)\} \qquad \text{(sifting theorem)}$$

$$= \frac{1}{4}\{60\sigma_x^6 + (N-1)\,6\sigma_x^6\}$$

$$= \frac{54+6N}{4}(\sigma_x^2)^3; \qquad \text{(A.16)}$$

where $\omega = 0$ was assumed going from the second line to the third line in order to account for the maximum variance contribution from the sum term.

Following the same steps for a complex signal, (A.2) becomes:

$$E\{|1\text{-}1/2\ \hat{D}_{ips}|^2\}_{COMPLEX} = [N+2]\,(\sigma_x^2)^3. \qquad \text{(A.17)}$$

The variance of the 1-1/2 D*ips* estimate is given by:

$$\mathrm{var}\ \{1\text{-}1/2\ \hat{D}_{ips}\} = E\{|1\text{-}1/2\ \hat{D}_{ips}|^2\} - [E\{1\text{-}1/2\ \hat{D}_{ips}\}]^2.$$

Note that the variance is simply equal to the second moment since the mean of the estimate is zero. Therefore, the variance of the 1-1/2 D_{ips} estimate is given by (A.16) for a real zero-mean independent Gaussian process. If the process is complex zero-mean independent Gaussian with independent real and imaginary parts, the estimate's variance is given by (A.17).

APPENDIX B: THEORETICAL 1-1/2 D_{ips} SIGNAL POWER

This Appendix first shows how the theoretical 1-1/2 D_{ips} spectrum is calculated for an input signal consisting of a single real valued sinusoid. The computation is then repeated for a complex valued sinusoid. Both calculations are made using a rectangular window in (2.24).

A. REAL VALUED SINUSOID CALCULATION

The real valued sinusoidal input signal is

$$x(n) = \cos(\theta n), \tag{B.1}$$

where,

$$\theta = 2\pi\frac{m}{N}, \tag{B.2}$$

is the digital frequency of the signal. Substituting (B.1) into (2.24) using a rectangular window yields:

$$\text{1-1/2 } D_{ips}(n,\omega) = \frac{1}{2}\sum_{k=0}^{N-1}\cos^2(\theta n)\,\{\cos[\theta(n-k)] + \cos[\theta(n+k)]\}\,exp(-j\omega k). \tag{B.3}$$

After applying the trigonometric cosine product identity,

$$\cos(A-B) + \cos(A+B) = 2\cos A\cos B, \tag{B.4}$$

and pulling the cosine cubed term out of the summation since it is not dependent on k, (B.3) becomes:

$$\text{1-1/2 } D_{ips}(n,\omega) = \cos^3(\theta n)\sum_{k=0}^{N-1}\cos(\theta k)\,exp(-j\omega k). \tag{B.5}$$

The summation portion of (B.5) is the Fourier transform of the sinusoidal signal. Evaluating the transform at $\omega = \theta$ and expressing θ as shown in (B.2), the Fourier transform can be expressed in terms of delta functions:

$$\sum_{k=0}^{N-1} \cos\left(2\pi \frac{m}{N} k\right) exp\left(-j2\pi \frac{m}{N} k\right) = \frac{N}{2} \{ \delta(\omega - m) + \delta[\omega - (N - m)] \} . \qquad \text{(B.6)}$$

With (B.2) and (B.6), (B.5) now becomes:

$$\text{1-1/2 D}_{\text{ips}}(n, \omega) = \frac{N}{2} \cos^3\left(2\pi \frac{m}{N} n\right) \{ \delta(\omega - m) + \delta[\omega - (N - m)] \} . \qquad \text{(B.7)}$$

B. COMPLEX VALUED SINUSOID CALCULATION

For a complex valued sinusoid input,

$$\begin{aligned} x(n) &= exp\left(j2\pi \frac{m}{N} n\right) \\ &= exp(j\theta n) , \end{aligned} \qquad \text{(B.8)}$$

the 1-1/2 D$_{\text{ips}}$ estimate calculated by (2.24) using a rectangular window is:

$$\text{1-1/2 D}_{\text{ips}}(n, \omega) = \frac{1}{2} \sum_{k=0}^{N-1} |exp(j\theta n)|^2 \{ exp[-j\theta(n-k)] + exp[j\theta(n+k)] \} \, exp(-j\omega k) \qquad \text{(B.9)}$$

After factoring out the exponential terms that are not a function of k, and recognizing that the magnitude squared term is simply equal to one, (B.9) reduces to:

$$\text{1-1/2 D}_{\text{ips}}(n, \omega) = \frac{1}{2} \{ exp[j\theta n] + exp[-j\theta n] \} \sum_{k=0}^{N-1} exp(j\theta k) \, exp(-j\omega k) . \qquad \text{(B.10)}$$

With the Euler identity,

$$\frac{1}{2} \{ exp[j\theta n] + exp[-j\theta n] \} = \cos(\theta n) , \qquad \text{(B.11)}$$

and (B.2), (B.10) becomes:

$$\text{1-1/2 D}_{\text{ips}}(n, \omega) = \cos\left(2\pi \frac{m}{N} n\right) \sum_{k=0}^{N-1} exp\left(j2\pi \frac{m}{N} k\right) exp(-j\omega k) . \qquad \text{(B.12)}$$

The summation term in (B.12) is the Fourier transform of a complex sinusoid which is expressible in terms of a delta function yielding:

$$1\text{-}1/2\ \mathrm{D}_{\mathrm{ips}}(n, \omega) = N\cos\left(2\pi\frac{m}{N}n\right)\delta(\omega - m)\,. \tag{B.13}$$

APPENDIX C: COMPUTER PROGRAMS

The four extrinsic Matlab functions contained in this Appendix are written for Matlab version 4.0. If the functions are used with earlier versions of Matlab, contour plots and mesh plots will not be oriented correctly.

```
%function [P,freqindex,timeindex] = one_half(data,wintype,winlen,step);

%This function will calculate the 1 1/2 Dips spectral surface. The 1 1/2 Dips
%surface characteristics are determined by the selection of window
%type (wintype), window length (winlen) and the distance that the window
%is moved through the data sequence (step). The magnitude of the positive
 %half of the 1 1/2 D spectral plane is returned in the “P” matrix.
%Outputs “timeindex” and “freqindex” aid in axis labeling.
%The inputs are:
%data - The input observations vector. The length should be a power of 2.
%wintype: ‘0’ Rectangular Window
%            ‘1’ Hamming Window
%winlen - The desired width of the window, normally half of the input
%            length.
%step - Time step desired, can be ‘1’ or a multiple of ‘2’

%prepared by Karen A. Hagerman, 06 May 1992.
%modified by Jeff McAloon, 01 June 1993.


function [P,freqindex,timeindex,svect]=one_half(data,wintype,winlen,step)

[datarows,datacolumns]=size(data);
if datarows~=1
   data=data.’;
end
siglen=length(data);
```

```
if wintype==0
    win=ones(winlen-1,1);
 elseif wintype==1
    win=hamming(winlen-1);
end
W=[win(winlen/2:-1:1)];
x=[zeros(1,winlen) data zeros(1,winlen)].';
p=zeros(siglen/step,winlen);


index=1;
for n=winlen+1:step:siglen+winlen-step+1
    xt=x;
    ll=max((winlen+1),(n-(winlen/2-1)));
   ul=min((n+(winlen/2-1)),(siglen+winlen));
    ctp=xt(ll:ul);
    mtp=mean(ctp);
    xt(ll:ul)=ctp-mtp;
    xdt=xt((n-(winlen/2-1)):(n+(winlen/2-1)));
    Xn=[abs(xt(n))^2;abs(xt(n))^2];
    Xm=[conj(xt(n:-1:n-(winlen/2-1))) xt(n:n+(winlen/2-1))];
    product=((Xm*Xn).*W).';
    product=[product 0 conj(product(winlen/2:-1:2))];
    pt=0.5*fftshift(fft(product));
    p(index,:)=pt;
   index=index+1;
end


P=abs(p(:,winlen/2+1:winlen));
```

```
[prow,pcolumn]=size(P);
freqindex=[0:pcolumn-1];
timeindex=[1:prow];
```

```
% function [P,freqaxis,timeaxis] = spectro(data,wintype,winlen);
%
% This function calculates spectrogram of the supplied sequence,
% “data”. The user must specify:
% “wintype” - “0” for a rectangular window.
%                   “1” for a Hamming window.
% “winlen” - The desired length of the data window.
% The time step is fixed at one and the spectrogram is calculated with
% non-normalized periodograms. The time-frequency surface is returned
% in the “P” matrix. The columns of “P” are the frequency bins while the
% rows are the time steps. Time-frequency axis labeling vectors,
% “freqaxis” and “timeaxis” are also returned to aid in the plotting
% of results.

% prepared by Jeff McAloon, 01 June 1993.


function [pow,freqaxis,timeaxis] = spectro(data,wintype,winlen)

x=data;
xlen=length(x);
N=winlen;
[row, col]=size(x);
if row > 1
   x=x.';
end
x=[zeros(1,N/2) x zeros(1,N/2)];
```

```
if wintype == 1
    win=hamming(N)';
elseif wintype == 0
    win=ones(1,N);
end

for ind=1:xlen
    xseg=x(ind:(N-1+ind));
    xseg=xseg-mean(xseg);
    SP=(abs(fft(xseg.*win,N))).^2;
    POW(ind,:)=fftshift(SP);
 end

pcw=POW(:,N/2+1:N);
timeaxis=0:xlen-1;
freqaxis=0:N/2-1;
```

```
% function [bis, baxis] = indbis(data,datseg,lag,tlen,win)

% This function computes the bispectrum of real and complex valued data
% sequences by the indirect method. The tri-correlation sequence is
% computed for a specified number of segments. These segment sequences
% are then averaged to obtain the final tri-correlation sequence. The
% three available tri-correlation window options are the unit hexagonal
% window, Parzen's window, and the optimum window (also known as Sazaki's
% minimum bias window). The required function call arguments are:
% "data" -> input data sequence vector.
% "datseg" -> number of samples to be used in each segment.
% "lag" -> number of lags to be used in the computation of the
%               tri-correlation sequence.
% "tlen" -> square dimension of the two dimensional FFT to be used
%               on the final form of the tri-correlation sequence.
% "win" -> "0" for unit hexagonal window
%               "1" for optimum window
%               "2" for Parzen's window.
% The function returns the complex bispectrum in the "tlen-by-tlen" sized
% matrix "bis". A vector "baxis" is returned that can be used to label
% both bispectrum frequency axes.

% prepared by Jeff McAloon, 01 Jun 1993.

function [bis, baxis] = indbis(data,datseg,lag,tlen,win)

% Initialize parameters and reorient data if necessary:
```

```
N=length(data);
Ns=fix(N/datseg);
[rd, cd]=size(data);
if cd > 1
    x=data.';
else
    x=data;
end
x=[0;x-mean(x)];
Rt=zeros(2*lag+1);
rx=zeros(2*lag+1);

% Compute tri-correlation sequence:
for i=1:Ns
    for j=1:datseg
        y(j)=x(j+(i-1)*datseg);
    end
    y=y-mean(y);
    for m=1:(2*lag+1)
        for n=1:(2*lag+1)
            s1=max([1,-(m-(lag+1)),-(n-(lag+1))]);
            s2=min([lag,lag-(m-(lag+1)),lag-(n-(lag+1))]);
            r=0;
            for k=s1:s2
                r=r+conj(y(k+1))*y(k+1+(m-(lag+1)))*y(k+1+(n-(lag+1)));
            end
            rx(n,m)=r;
        end
    end
    Rt=Rt+rx;
    rx=[];
end
R=flipud(Rt/Ns);
R=Rt/Ns;

% Determine tri-correlation window function:
if win == 0                                       % unit hexagonal window
    W=ones(size(R));
elseif win == 1                                   % optimum window
    itm=0;
    for m=-lag:lag
       itm=itm+1;
       wm=abs(sin(pi*m/lag))/pi + (1-abs(m/lag))*cos(pi*m/lag);
       itn=0;
       for n=-lag:lag
           itn=itn+1;
           wn=abs(sin(pi*n/lag))/pi + (1-abs(n/lag))*cos(pi*n/lag);
           wmn=abs(sin(pi*(m-n)/lag))/pi + (1-abs((m-n)/lag))*cos(pi*(m-n)/lag);
           W(itm,itn)=wm*wn*wmn;
       end
    end
elseif win == 2                                   % Parzen's window
    itm=0;
    for m=-lag:lag
       itm=itm+1;
       if abs(m) <= lag/2
```

```
           wm=1-6*(abs(m)/lag)^2+6*(abs(m)/lag)^3;
     else
         wm=2*(1-abs(m)/lag)^3;
    end
    itn=0;
    for n=-lag:lag
        itn=itn+1;
        if abs(n) <= lag/2
            wn=1-6*(abs(n)/lag)^2+6*(abs(n)/lag)^3;
            wmn=1-6*(abs(m-n)/lag)^2+6*(abs(m-n)/lag)^3;
        else
            wn=2*(1-abs(n)/lag)^3;
            wmn=2*(1-abs(m-n)/lag)^3;
        end
        W(itm,itn)=wm*wn*wmn;
      end
  end
end

% Compute bispectrum.
bis=fftshift(fft2(R.*W,tlen,tlen));
baxis=(-tlen/2):(tlen/2-1);
```

```
% function Amtrx = gen_a(a1_elem);

% This function computes the parameter vectors needed to compute a fourth
% order moment slice in the IHOMS time-frequency method implemented by
% the extrinsic matlab function ATH_IMS.M. The first element, a1, of
% each IHOMS parameter vector is specified in the function call as the
% vector, “a1_elem”. The length of “a1_elem” is the number of distinct
% parameter vector sets (i.e.one set would be [1 a1 a2 a3]) returned as
% columns in the matrix “Amtrx”. This also equals the number of time-
% frequency surfaces summed by IHOMS to form the final time-frequency
% representation.

% prepared by Jeff McAloon 01 June 1993.

function Amtrx = gen_a(a1_elem)

% Verify input vector a column vector:
[rin, cin]=size(a1_elem);
if cin > 1
   a1v=a1_elem.’;
end

% Iteratively solve for parameter vectors:
A=[];
for p=1:length(a1v)
   a3v=-5:1:5;
   a2a=100; a2b=1; i=1; md=99;
```

```
    while (abs(a2b-a2a) >= 0.1) & (i <= length(a3v))
      a2a=1-a1v(p)-a3v(i);
      a2b=real(sqrt(1-a1v(p)^2-a3v(i)^2));
      if abs(a2a-a2b) <= md
        indmin=i;
      end
      md=min(md,abs(a2a-a2b));
      id=max(2,indmin);
      i=i+1;
    end
    a3v=a3v(id-1):0.0001:a3v(id);
    a2a=1; a2b=2; i=0;
    while (abs(a2b-a2a) >= 0.0001) & (i <= (length(a3v)-1))
      i=i+1;
      a2a=1-a1v(p)-a3v(i);
      a2b=sqrt(1-a1v(p)^2-a3v(i)^2);
    end
    a1=a1v(p); a2=a2a; a3=a3v(i);
    a=[1 a1 a2 a3];
    A=[A, a.'];
end

Amtrx=A;
```

```
% function surf = ath_ims(M,u,b,N,SNR,A,g)

% This function computes the IHOMS time-frequency representation of
% multi-component chirp signals in white Gaussian noise. Input arguments
% are:
% M -> number of chirp components.
% u -> vector of chirp starting frequencies.
% b -> vector of halved chirp slew rates.
% N -> data record length.
% SNR -> desired SNR. If noise free signal desired enter "99".
% A -> matrix whose columns are the "a" parameter vectors needed to
%        form the moment slice. See extrinsic Matlab function GEN_A
%        generate this matrix.
% g -> kernel function parameter.
% The moment slice is computed using 100 lags. The IHOMS surface is
% plotted along with a 1-D radial maxima plot that aids in locating and
% characterizing the chirps present in the signal. In addition, the
% function returns a 100 X 100 matrix, "surf". The columns are time/lag
% and the rows are frequency bins.

% prepared by Jeff McAloon, 01 June 1993.

function surf = ath_ims(M,u,b,N,SNR,A,g)

% Initialize parameters:
randn('seed',0)
if SNR == 99                              % noise free
```

```
    GG=0;
else
    GG=1;
end
Z=sqrt(10^(GG*SNR/10));
maxlag=100;
j=sqrt(-1);
[rA,cA]=size(A);
SURF=zeros(100);

% IHOMS algorithm:
for k=1:cA
 a=A(:,k);
 kn=k
 TF=[];
 for n=0:maxlag-1
    for lag=0:maxlag-1
       no=GG*randn(1,length(a));
       term(1)=0;
       for p=1:M
          term(1)=term(1)+Z*exp(j*2*pi*(((lag)/N)*u(p)+(((lag)/N)^2)*b(p)));
       end
       term(1)=term(1)+no(1);
       for i=2:length(a)
          arg=n+a(i)*lag;
          if arg < 0
             term(i)=0;
          else
```

```
               term(i)=0;
               for q=1:M
                  term(i)=term(i)+Z*exp(j*2*pi*(((arg)/N)*u(q)+(((arg)/N)^2)*b(q)));
               end
               term(i)=term(i)+no(i);
            end
         end
         kern=exp(-g*lag^2);
         m(lag+1)=kern*conj(term(1))*term(2)*term(3)*term(4);
      end
      TF(n+1,:)=fft(m);
   end
   SURF=SURF+TF;
end

% Orient matrix for MATLAB 4.0 contour plot:
surf=rot90(flipud(abs(SURF)),-1);

% Plot IHOMS surface:
t=0:maxlag-1;
f=0:maxlag-1;
subplot(121),contour(t,f,surf,2),title(‘IHOMS surface’), ...
xlabel(‘time/lag’),ylabel(‘frequency’)

% IHOMS 1_D search routine and plot:
theta=0:0.01:pi/2;
for in=1:length(theta)
   D(in)=0;
```

```
    for r=30:90
        t=1+r*cos(theta(in));
        f=1+r*sin(theta(in));
        D(in)=D(in)+surf(f,t);
     end
end
subplot(122),plot(theta,D),xlabel('theta in radians'), ...
ylabel('D(theta)'),title('1-D radial maxima plot')
```

REFERENCES

1. Nikias, C.L., and Raghuveer, M.R., "Bispectrum Estimation: A Digital Signal Processing Framework," *Proceedings of the IEEE*, Vol. 75, pp. 869-891, 1987.

2. Kay, S. M., *Modern Spectral Estimation*, Prentice-Hall, Englewood Cliffs, New Jersey, 1988.

3. Mendel, J. M., "Tutorial on Higher-Order Statistics (Spectra) in Signal Processing and System Theory: Theoretical Results and Some Applications," *Proceedings of the IEEE*, Vol. 79, pp. 278-305, 1991.

4. Haykin, S., editor. *Advances in Spectrum Analysis and Array Processing*, Vol. 1, Prentice-Hall, Englewood Cliffs, New Jersey, 1991.

5. Hagerman, K. A., *Instantaneous Power Spectrum and 1-1/2 D Instantaneous Power Spectrum Techniques*, Master's Thesis, Naval Postgraduate School, Monterey, California, June 1992.

6. Petropulu, A. P., "Detection of Multiple Chirp Signals based on a Slice of the Instantaneous Higher-Order Moments," *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pp. 261-264, IEEE Press, Victoria, B.C., Canada, Catalog No. 92TH0478-8, October 1992.

7. Whalen, A. D., *Detection of Signals in Noise*, Academic Press, San Diego, California, 1971.

8. Wolinsky, M.A., *Invitation to the Bispectrum, Tech. Report No. ARL-TR-88-7*, Applied Research Laboratories, The University of Texas at Austin, August 1988.

9. Jouny, I.I., and Moses, R.L., "The Bispectrum of Complex Signals: Definitions and Properties," *IEEE Transactions on Signal Processing*, Vol. 40, No. 11, pp. 2833-2836, November 1992.

10. Swami, A. and Mendel, J. M., "Cumulant-Based Approach to Harmonic Retrieval and Related Problems," *IEEE Transactions on Signal Processing*, Vol. 39, No. 5, pp. 1099-1109, May 1991.

11. Therrien, C. W., *Discrete Random Signals and Statistical Signal Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, 1992.

12. Kletter, D., and Messer, H., "Suboptimal Detection of Non-Gaussian Signals by Third-order Spectral Analysis," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* Vol. 38, No. 6, pp. 901-909, June 1990.

13. Hinich, M. J., "Detecting a Transient Signal by Bispectral Analysis," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* Vol. 38, No. 7, pp. 1277-1283, July 1990.

14. Hi-Spec Software Reference Manual for Signal Processing with Higher-Order Spectra, United Signals and Systems, Inc., 1990.

15. Sasaki, K., and Yamashita, Y., "Minimum Bias Windows for Bispectral Estimation," *Journal of Sound and Vibration,* Vol. 40, No. 1, pp. 139-148, 1975.

16. Hippenstiel, R. D. and De Oliveira, P. M., "Time-Varying Spectral Estimation Using the Instantaneous Power Spectrum (IPS)," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* Vol. 38, No. 10, pp. 1752-1759, October 1990.

17. Ross, S., *A First Course in Probability,* MacMillian Publishing Co., New York, New York, 1988.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
Cameron Station
Alexandria, VA 22304-6145

2. Dudley Knox Library, Code 52 2
Naval Postgraduate School
Monterey, CA 93943-5002

3. Chairman, Code EC 1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5121

4. Prof. Ralph Hippenstiel, Code EC/Hi 2
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5121

5. Prof. Monique P. Fargues, Code EC/Fa 1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5121

6. Prof. R. Clark Robertson, Code EC/Rc 1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5121

7. Naval Command, Control, and Ocean Surveillance Center 1
NRAD
Attn: Dr. C.E. Persons (Code 732)
San Diego, CA 92152

8. Commanding Officer 2
Surface Warfare Officers School Command
Attn: Lt. Jeff McAloon
Newport, RI 02841